Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1989

Parameter plane design method.

Shin, Dong Ryong.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/26148

Downloaded from NPS Archive: Calhoun

DUDLEY
KNOX
LIBRARY

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

S4741

PARAMETER PLANE DESIGN METHOD

by

Dong Ryong Shin

March 1989

Thesis Advisor George J. Thaler

Approved for public release; distribution is unlimited.

T242351

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

lassified

ty classification of this page

REPORT DOCUMENTATION PAGE

eport Security Classification Unclassified	1b Restrictive Markings
ecurity Classification Authority eclassification Downgrading Schedule	3 Distribution Availability of Report Approved for public release; distribution is unlimited.
rforming Organization Report Number(s)	5 Monitoring Organization Report Number(s)

ame of Performing Organization al Postgraduate School	6b Office Symbol *(if applicable)* 62	7a Name of Monitoring Organization Naval Postgraduate School
ddress *(city, state, and ZIP code)* nterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
ame of Funding Sponsoring Organization	8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number

ddress *(city, state, and ZIP code)*	10 Source of Funding Numbers			
	Program Element No	Project No	Task No	Work Unit Accession No

itle *(include security classification)* PARAMETER PLANE DESIGN METHOD

ersonal Author(s) Dong Ryong Shin

Type of Report ster's Thesis	13b Time Covered From To	14 Date of Report *(year, month, day)* March 1989	15 Page Count 104

upplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or po-
on of the Department of Defense or the U.S. Government.

osati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)* Parameter, Integral-Square error, Cost Function
	Group	Subgroup	

bstract *(continue on reverse if necessary and identify by block number)*

In this thesis a control systems analysis package is developed using parameter plane methods. It is an interactive, user-
ndly computer aid to plot families of performance index cost curves. By superimposing the cost curves on the parameter
ne curves the designer is able to choose values of the parameters which provide a good compromise between cost and
amic behavior.

istribution Availability of Abstract nclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
Name of Responsible Individual orge J. Thaler	22b Telephone *(include Area code)* (408) 626-2134	22c Office Symbol 62Tr

FORM 1473,84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Parameter Plane Design Method

by

Dong Ryong Shin
Lieutenant Commander, Korean Navy
B.S., Korean Naval Academy, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

ABSTRACT

In this thesis a control systems analysis package is developed using parameter plane methods. It is an interactive, user-friendly computer aid to plot families of performance index cost curves. By superimposing the cost curves on the parameter plane curves the designer is able to choose values of the parameters which provide a good compromise between cost and dynamic behavior.

Thesis
S4741
C.1

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND THEORY AND EQUATIONS 3
  A. FUNDAMENTAL CONCEPT 3
  B. PARSEVAL'S THEOREM 4
  C. MINIMIZATION PROCEDURE 6

III. PERFORMANCE INDEX CURVE EVALUATION 8
  A. BACKGROUND 8
    1. Overview 8
    2. Software / Hardware 8
    3. Two Dimensional Contouring 9
  B. THE PROGRAM 11
    1. Subroutine SIZE 12
    2. Subroutine LDFILE 13
    3. Subroutine MONPRT 13
  C. ANSI MODULE 16

IV. PARAMETER PLANE DESIGN -GRAPHICAL METHOD 18
  A. OVERVIEW 18
  B. GRAPHICAL SOLUTIONS 19
    1. Example 4-1(cascade compensator) 20
    2. Example 4-2 (feedback compensator) 32
    3. Example 4-3 (optimization of controller) 43

V. CONCLUSIONS AND RECOMMENDATIONS 52
  A. CONCLUSIONS 52
  B. RECOMMENDATIONS 52

APPENDIX A. TABULATED VALUES OF THE INTEGRAL FORM 53

MONTEREY, CALIFORNIA

APPENDIX B. PERFORMANCE INDEX PROGRAM. 58

APPENDIX C. D.S.L SIMULATION PROGRAM AND DATA FILE. 68
- A. SIMULATION PROGRAM FOR EXAMPLE 4-1. 68
- B. DATA FILE FOR EXAMPLE 4-1. 72
- C. SIMULATION PROGRAM FOR EXAMPLE 4-2. 75
- D. DATA FILE FOR EXAMPLE 4-2. 79
- E. SIMULATION PROGRAM FOR EXAMPLE 4-3. 82
- F. DATA FILE FOR EXAMPLE 4-3. 85

APPENDIX D. USER'S MANUAL 87

LIST OF REFERENCES 91

INITIAL DISTRIBUTION LIST 92

LIST OF TABLES

Table 1. PARAMETER TABLE 12
Table 2. SIMULATION RESULTS. 32
Table 3. SIMULATION RESULTS. 43
Table 4. REID RESULTS. 44

LIST OF FIGURES

Figure 1. Block diagram of feedback control system. 3
Figure 2. The contour grid two-dimensional real array z from Ref.3. 9
Figure 3. Performance Index Program Flow Chart. 11
Figure 4. The plot size subroutine. 12
Figure 5. The data load subroutine. 13
Figure 6. The printer/output menu. 14
Figure 7. The Hardware Interface Menu. 15
Figure 8. Baud (data transfer) Rate Menu. 15
Figure 9. Parity Menu. 16
Figure 10. Graphical analysis path. 19
Figure 11. The third order cascade compensated system. 20
Figure 12. Combine zeta and omega curves for example 4.1 21
Figure 13. The performance index curves for $\int E^2 dt$. 24
Figure 14. The performance index curves for $\int |E| \, t \, dt$. 25
Figure 15. The performance index curves for $\int |E| \, dt$. 26
Figure 16. Combine zeta, omega and cost curves for $\int E^2 dt$. 28
Figure 17. Combine zeta, omega and cost curves for $\int |E| \, t \, dt$. 29
Figure 18. Combine zeta, omega and cost curves for $\int |E| \, dt$. 30
Figure 19. Step response on example 4.1 31
Figure 20. Velocity and acceleration feedback compensated system. 32
Figure 21. Constant zeta and omega curves for example 4.2 34
Figure 22. The performance index curves for $\int E^2 dt$. 35
Figure 23. The performance index curves for $\int |E| \, t \, dt$. 36
Figure 24. The performance index curves for $\int |E| \, dt$. 37
Figure 25. Combine zeta, omega and cost curves for $\int E^2 dt$. 39
Figure 26. Combine zeta, omega and cost curves for $\int |E| \, t \, dt$. 40
Figure 27. Combine zeta, omega and cost curves for $\int |E| \, dt$. 41
Figure 28. Step response on example 4.2 42
Figure 29. Optimization of controller. 43
Figure 30. Constant zeta curves for example 4.3 45
Figure 31. Rearrangement of block diagram. 46

Figure 32. The cost curves for $\frac{1}{T}\int_0^T(\lambda\psi_e^2 + \delta^2)\,dt$. 47
Figure 33. Combine zeta and cost curves for $\frac{1}{T}\int_0^T(\lambda\psi_e^2 + \delta^2)\,dt$. 49
Figure 34. The yaw response vs time. 50
Figure 35. The rudder activity vs time. 51

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to my thesis advisor, Professor George J.Thaler, for his invaluable efforts and patience in assisting me through my research. I also want to thank my wife Sook Hee and my children, Yoon Hyung and Han Na for their support and patience away from our home during two and half years in the United States of America.

Finally, I would also like acknowledge the Republic of Korean Navy for the opportunity to attend the Naval Postgraduate School.

I. INTRODUCTION

When analyzing or designing a control system, it is important to study the effects on overall system performance of varying one or more parameters (mass, inertia, gain, resistance, etc). It is equally important to determine whether a desired dynamic behavior can be achieved with any set of values for the parameters ... if not, redesign is indicated.

The analysis and synthesis of linear feedback control systems, or the compensation of same, can be realized by parameter plane methods. The parameter plane method, which works well for two variable parameters and which may be extended to three or more parameters, is purely algebraic, and the resulting plots are valuable aids to analysis.

Furthermore, the performance index cost curves in terms of integral-square error and other cost functions can be superimposed on the parameter plane graphically. That is very useful for analysis and for designing a control system when its characteristic equation has two parameters. Since the coefficients of the characteristic polynomial are determined by the system parameters, it follows that some relationship exists between the value of any parameter and the value of the characteristic roots. The integral-square error as a performance index is also determined by the system parameters.

Let us examine the choice of the integral-square error as a performance index. In many applications the performance of feedback control systems can be considered either satisfactory if the errors are below a certain limit, or unsatisfactory, if the errors are above a certain limit. The hypothesis that the configuration is fixed refers to the form of the system; it is assumed that one or more parameters can be adjusted. Given the input and ideal output signals as transient functions of time and given the configuration of the control system, find the values that the free parameters should have in order to minimize the integral-square error.

One approach to solving this problem starts with the Fourier transforms of the input and ideal output signals. The integral-square error is then easily formulated by means of Parseval's theorem. If the transform of the error is a rational function of the complex frequency, the integral-square error can be easily evaluated by means of a table of integral, which is in Appendix A. When the integral-square error is evaluated, it appears as a function of the free parameters.

The object of this thesis, then, is to develop a user friendly interactive computer program to plot families of performance index cost curves. A secondary objective is to demonstrate the use of these curves in the design of the system. The design procedure starts with the performance index cost curves superimposed on parameter plane curves. The result of the analysis is a choice of values for the parameters such that the system performance is the best available compromise between transient performance and cost.

II. BACKGROUND THEORY AND EQUATIONS

A. FUNDAMENTAL CONCEPT

Classical control theory, historically speaking, has been applied to the design problem with the assistance of graphical presentation and trial and error methods. Such methods have been quite successful in the development of good control systems, but do not answer the question "Is this the best system possible." The engineer and mathematician have always looked for the "best" control system. Early methods used "standard forms." The concept of a "performance index" was invoked, also at an early point, for, if the chosen index could be satisfied, one could say that the resulting system was indeed the "best" as measured by the index.

Let us take the performance index that is the integral-square value of the error between the ideal output and the actual output. A simple feedback control system with adjustable parameters is assumed as presented in Figure 1.

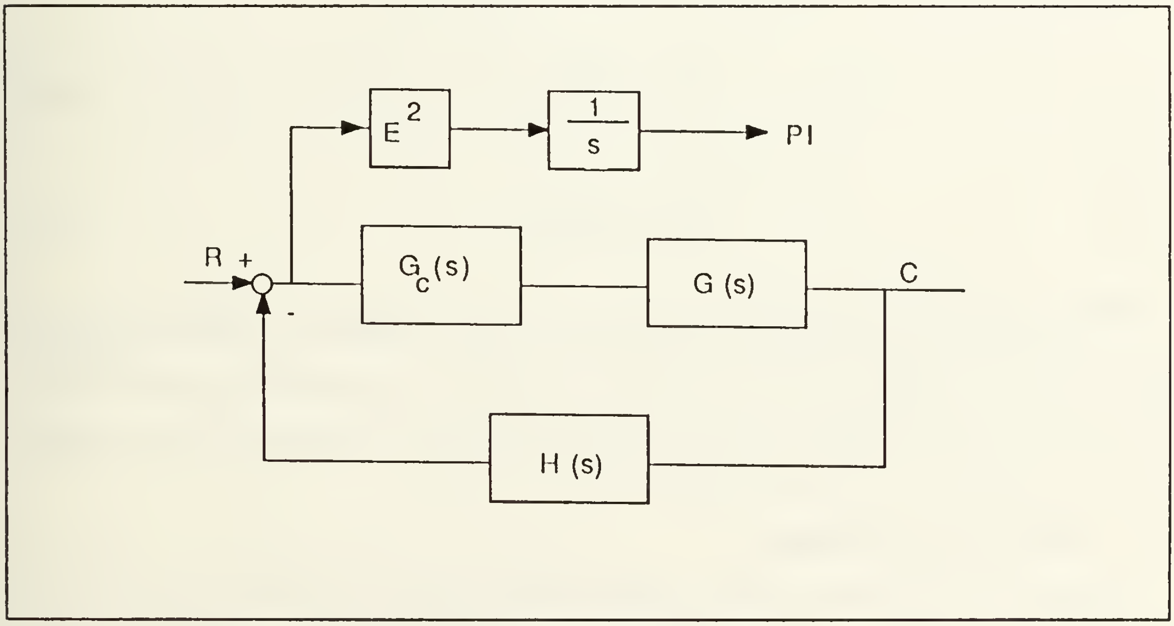


Figure 1. Block diagram of feedback control system.

The performance index in terms of integral-square error is defined as:

$$\mathbf{PI} = \int E^2 \, dt \tag{2.1}$$

where **PI** ; performance index

E ; input - actual output

and the criterion was to design the system such that this index was minimized. For linear systems this problem can be solved by applying Parseval's theorem [Ref. 1: p. 43] to obtain explicit relationships for system parameters. One should note that a system designed to minimize this performance index has a step response with considerable overshoot and oscillation, and this may not be acceptable for some applications, but where such dynamic characteristics are suitable the design is straightforward. Other performance indices which have been found useful are

$$\mathbf{PI} = \int |E| \, t \, dt \tag{2.2}$$

$$\mathbf{PI} = \int |E| \, dt \tag{2.3}$$

Both of these are nonlinear and do not have analytic solutions. Designs using such criteria were accomplished using simulation. The solution of an optimal control problem is the result of minimizing the chosen cost function and is the design which best satisfies the criteria defined by that cost function.

B. PARSEVAL'S THEOREM

Suppose we wish to know the area beneath the product of two functions $x_1(t)$ and $x_2(t)$ of the infinite interval of the time from minus infinity to plus infinity. Let this area be denoted by the symbol **I** (which stands for integral), thus

$$\mathbf{I} = \int_{-\infty}^{+\infty} x_1(t) \, x_2(t) \, dt \tag{2.4}$$

Suppose, in addition, that the time functions are Fourier transformable and have transforms $x_1(s)$ and $x_2(s)$, respectively. We wish to express the integral **I** directly in terms of these transforms in order to obviate inverse transforming $x_1(s)$ and $x_2(s)$ when these are given rather than the time functions themselves. To do this we observe that

$$x_2(t) = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} e^{st} x_2(s)\, ds \tag{2.5}$$

Substitution of this value of $x_2(t)$ into Equation (2.4) yields

$$\mathbf{I} = \int_{-\infty}^{+\infty} x_1(t) \left[\frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} e^{st} x_2(s)\, ds \right] dt \tag{2.6}$$

Let us now interchange the order of the integrations so that we integrate with respect to time first

$$\mathbf{I} = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} x_2(s)\, ds \int_{-\infty}^{+\infty} e^{st} x_1(t)\, dt \tag{2.7}$$

By the direct Fourier transform we know that

$$x_1(s) = \int_{-\infty}^{+\infty} e^{-st} x_1(t)\, dt \tag{2.8}$$

Thus the integral with respect to time on the right side of Equation (2.7) may be evaluated as

$$\int_{-\infty}^{+\infty} e^{st} x_1(t)\, dt = x_1(-s) \tag{2.9}$$

This permits us to write Equation (2.7) as

$$\mathbf{I} = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} x_1(-s)\, x_2(s)\, ds \tag{2.10}$$

This is the desired expression for the integral **I**.

An important particular case of the above result occur when the two time functions involved are identical so that the integral I becomes

$$\mathbf{I} = \int_{-\infty}^{+\infty} x^2(t)\, dt \tag{2.11}$$

We call this integral the integral-square value of the time function $x(t)$.

By Equation (2.10) we know this can be expressed as

$$\mathbf{I} = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} x(-s)\, x(s)\, ds \tag{2.12}$$

This result is known as Parseval's theorem. This theorem represents a very convenient way of expressing the integral-square value of a time function in terms of its transform.

C. MINIMIZATION PROCEDURE

On the basis of Parseval's theorem we are now in a position to formulate the procedure for minimizing the integral-square error for transient input signals by adjustment of the free parameters.

This procedure involves four steps. The first step is to express the Fourier transform of the error as a function of the complex frequency s. This function will involve the free parameters of the system as unknown coefficients. The second step is to express the integral-square error $\mathbf{I}_e$ in terms of the error transform **E(s)** by Parseval's theorem, i.e.,

$$\mathbf{I}_e = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} E(-s)\, E(s)\, ds \tag{2.13}$$

At this stage, provided **E(s)** is a rational function, the integral-square error will appear in the form

$$\mathbf{I}_e = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} \frac{c(-s)\, c(s)}{d(-s)\, d(s)} \tag{2.14}$$

where $c(s)$ and $d(s)$ are polynomials in s. The third step is to evaluate the integral. Fortunately, definite integrals of this form have been evaluated in terms of the coefficients appearing in the polynomials. A table of these integrals along with an explanation of how they were evaluated is given in Appendix A. Upon carrying out the integration we have the integral-square error expressed as a function of the free parameters p_1 through p_k. Symbolically the integral-square error at this stage is expressed as

$$I_e(p_1, p_2, \cdots, p_k)$$

The fourth step of the minimization procedure is to adjust the values of the free parameters in such a way as to minimize the integral-square error. Formally this can be done in the standard manner of equating the partial derivatives of the integral-square error to zero and solving the resultant set of equations for the values of the parameters. In symbols this means solving the set of k equations of the form

$$\frac{\partial I_e}{\partial p_k} = 0$$

Unfortunately in many practical problems this formal procedure for finding the parameter values that minimize the integral-square error leads to sets of non-linear equations in the parameters which are so complex that straightforward methods of solution do not exist.

If the equations for the parameters obtained by setting the partial derivatives of the integral-square error with respect to the parameters equal to zero are too complex to be solved by analytical methods, we may resort to trial-and-error or successive approximation techniques. However, if such methods are necessary to solve the equations for the parameters, it is probably best to go back to the expression for the integral-square error itself and plot this as a function of one of the parameters, holding the other parameters fixed. By using a sufficient number of fixed values for these other parameters and making a sufficient number of plots, values can be found for all the parameters that define the minimum value of the integral-square error closely enough for engineering purposes. By working directly with the integral-square error expression, we avoid taking the partial derivative of this expression with respect to each of the free parameters. Taking these derivatives may be a very difficult and time-consuming chore.

III. PERFORMANCE INDEX CURVE EVALUATION

A. BACKGROUND

1. Overview

The performance index program is an interactive personal computer program whenever possible. Before using this program, we have to generate a datafile containing α , β and PI. This file will enable us to plot families of performance index cost curves. There currently exists, at Naval Postgraduate School, a DSL(Dynamic Simulation Language) [Ref. 2] program residing on the school's IBM 370 mainframe. By using the DSL program, we can easily obtain the above mentioned data file, and then, down load this file from the mainframe computer to floppy diskette. This data, now on floppy diskette, is to be used, along with the computer program developed in this research paper, to plot performance index curves. With this program, the user can examine particular values of the performance index curves. Options, such as printer type and size of output, can be selected by the user.

2. Software / Hardware

The program language used in coding this program is Microsoft FORTRAN77 V3.20. The Plotworks PLOT88 graphics library [Ref. 3] is used to generate output plots. The plotting of performance index curves conducted for this thesis were performed on an IBM-AT or IBM compatible 80286 machine. Due to the universality of FORTRAN coding, this program could be implemented on any machine capable of being programmed in FORTRAN. The source code of generating curves of constant performance index is listed in Appendix B.

Before running the program, ANSI.SYS should be incorporated in the personal computer's CONFIG.SYS file prior to running the performance index program. Through the ANSI.SYS device driver, system calls to clear the screen and position the cursor are enabled. If not, screen readability suffers although the program is still fully functional. More will be said on this in the ANSI module section of this chapter.

3. Two Dimensional Contouring

The control grid is constructed as follows Figure 2.

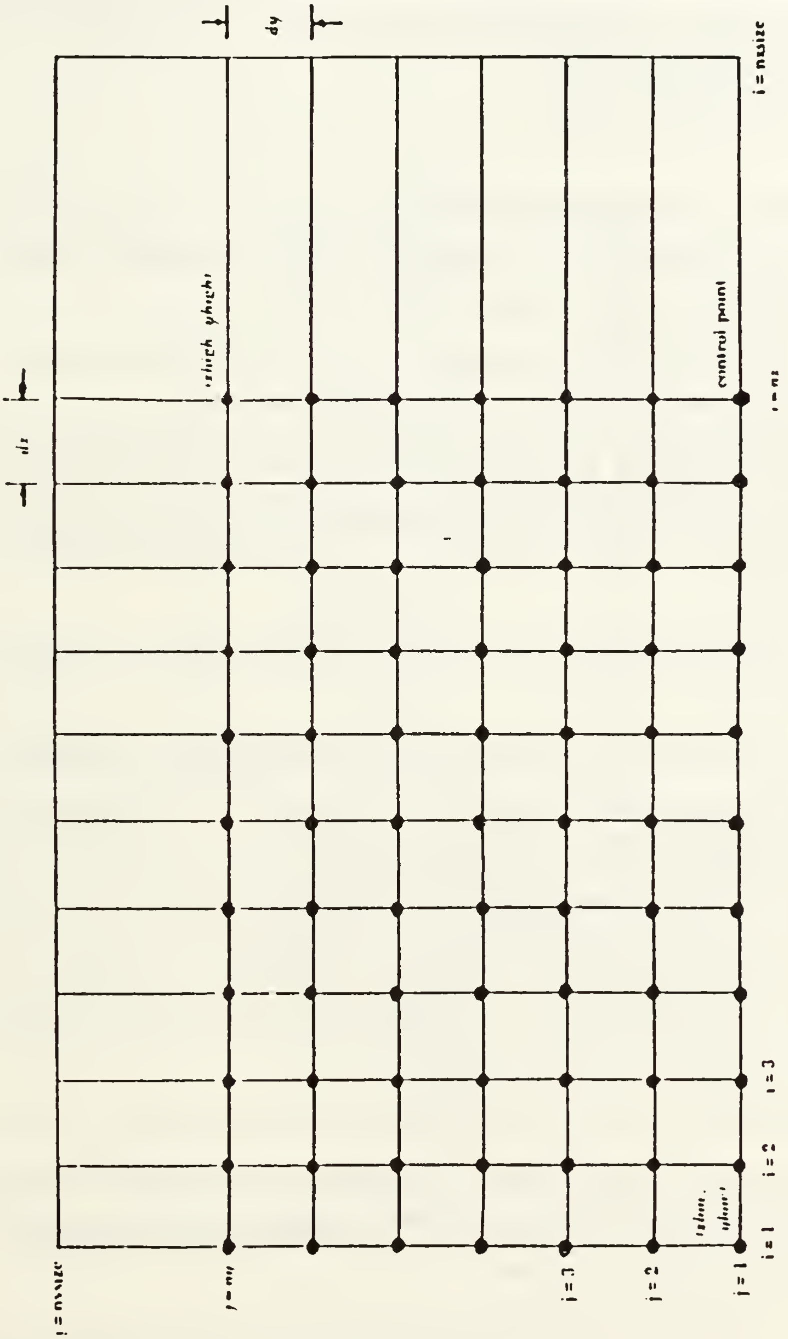


Figure 2. The contour grid two-dimensional real array z from Ref.3.

where • = dimensional size of real array z.

(nx.ny) = maximum index of area used.

(xlow.ylow) = lower left corner of the grid area in the world units.

(xhigh.yhigh) = upper right corner of the grid area in the world units.

(nxsize.nysize) = grid interval in world units.

dx = (xhigh-xlow)/(nx-1)

dy = (yhigh-ylow)/(ny-1)

Note that only the control points in the range (i = 1,.....,nx ; j = 1,.....,ny) are used in the array z. By convention, the first index is referred to as the row index and the second index is referred to as the column index.

The following sequence produces a two-dimensional contour map:

- A grid constructed over the surface Figure 2.
  At each control point of the grid, the z ordinate is determined. Using Laplace or Spline interpolation or a combination of the two, curves of constant z are constructed by PLOT88 graphics library. To do that, we can use the subroutine named ZGRID which is the PLOT88 library, this subroutine interpolates a set of arbitrarily spaced data points $xp(k)$, $yp(k)$, $k = 1$, $npts$ onto a rectangular grid $z(i, j)$ of arbitrary size.
- The grid control points may be smoothed. The ZSMTH subroutine which is the PLOT88 library smooths the gridded control points by applying Laplacian smoothing.
- Potions of the contour may be removed to make room for annotation.
- The contours are drawn. Each set of 4 grid points can be subdivided further into 4 or 16 subgrids for more precise contouring. In this program, the set of 4 grid points is subdivided into 16 subgrids. Variable contour line types and widths can be drawn. To make drawing more precisely, we can use the subroutine named ZCSEG which is the PLOT88 library. This subroutine converts the grid of control points into a set of contours and draws the contours. In addition, the capability of dividing each grid square into 1, 4, 16 subsections using cubic polynomial interpolation. This step may be needed if contours are close together or if the line quality of contour need improvement.

The size of the contour grid is defined in the main program. For most applications, a size of 50 × 50 control points is sufficient. The maximum number of control points is limited by the available space in the data segment of the program. A maximum of the 51 control levels can be drawn.

B. THE PROGRAM

The performance index program given in Appendix B itself consists of main program and six subroutines which make up the performance index package. Figure 3 is a flow chart illustrating the organization.

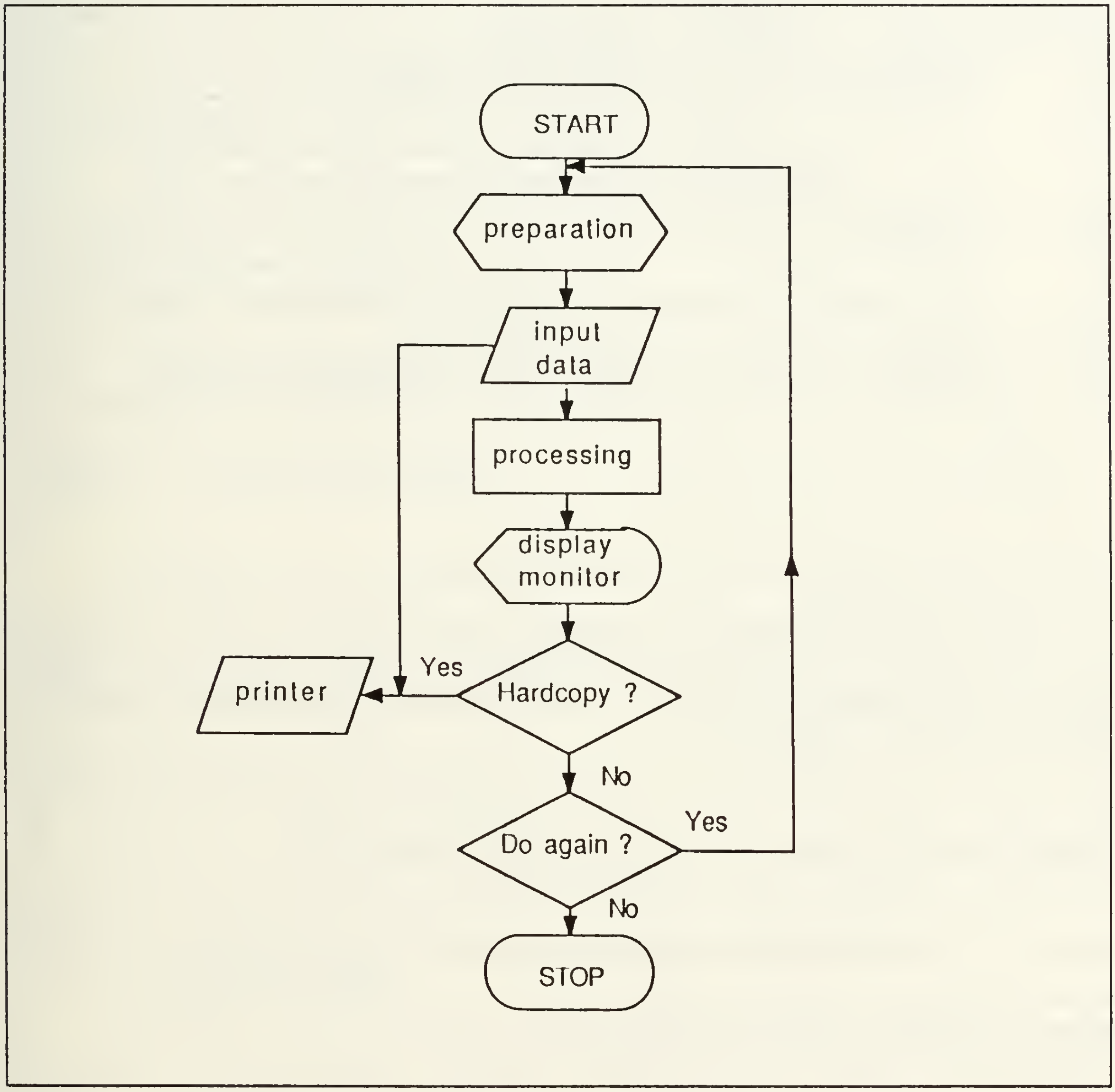


Figure 3. Performance Index Program Flow Chart.

When the program is initiated, the following questions are presented:

1. How many constant PI curves # = = = >

- The user has to enter an integer value which is must be less than 51.

2. Enter PI values, lwgt, ldig # = = = >

- Table 1 give us proper values.

Table 1. PARAMETER TABLE

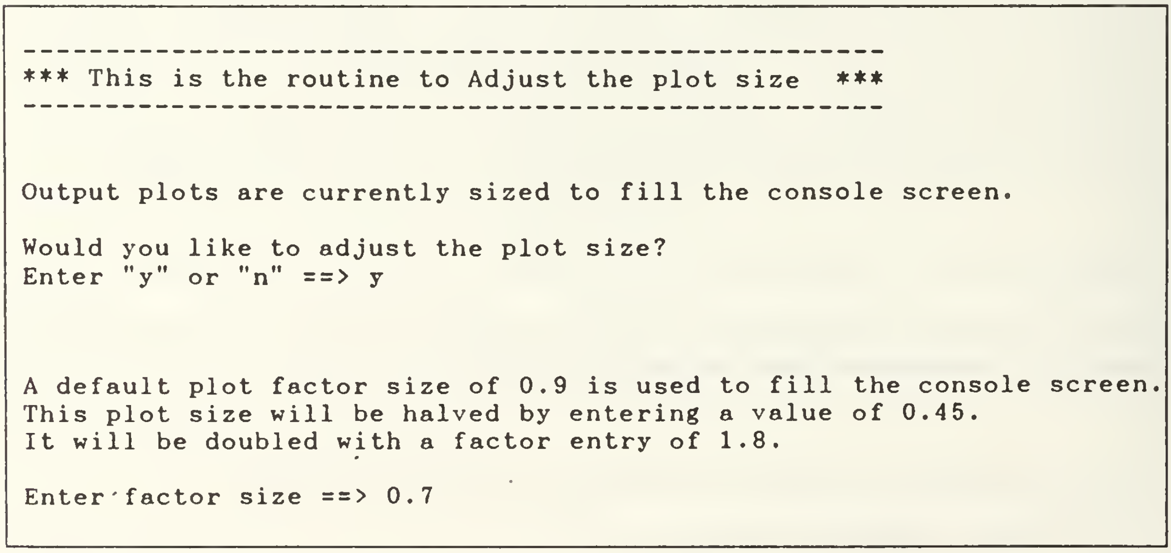
parameter	type	value	Description
PI	real		Performance Index
lwgt	integer	1 to 27	the line type for each PI curve the PI curve is drawn with a solid line. See PLOT88 Manual.[Ref. 3: p. 32]
ldit	integer	-1 to -27	the PI curves is drawn with a dashed line.
ldit	integer	> 0	the annotation control information for each PI. number of digits after decimal point.
ldit	integer	-1	omit decimal point.
ldit	integer	≤ -2	omit contour annotation for this level.

3. Enter number of data file points = = = >

- The user has to enter total number of data points.

1. Subroutine SIZE

The subroutine SIZE Figure 4 is displayed. This subroutine permits adjustment of plot size to both the screen and the printer. This option was utilized to scale plot outputs for inclusion in this thesis.

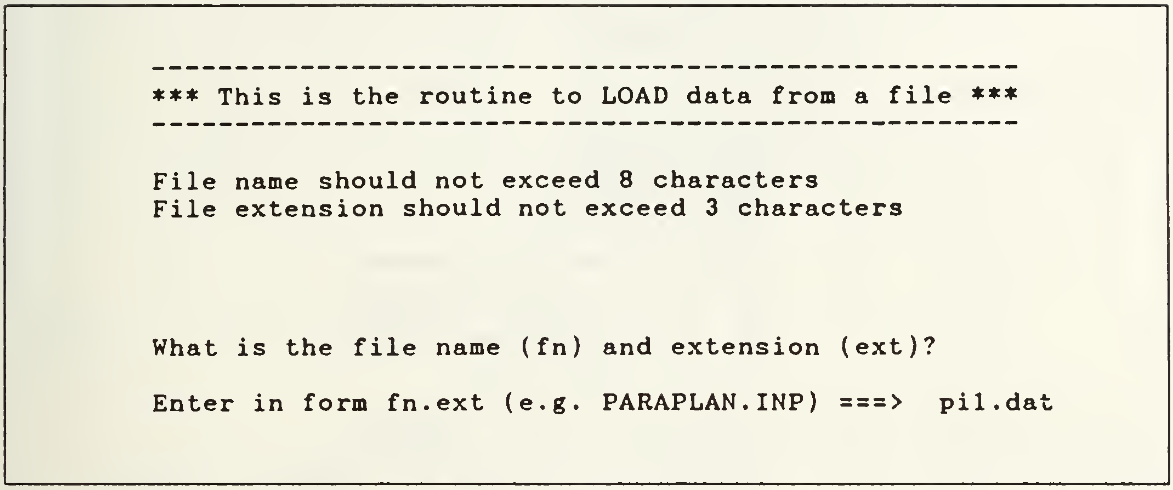


```
-----------------------------------------------------
*** This is the routine to Adjust the plot size  ***
-----------------------------------------------------


Output plots are currently sized to fill the console screen.

Would you like to adjust the plot size?
Enter "y" or "n" ==> y



A default plot factor size of 0.9 is used to fill the console screen.
This plot size will be halved by entering a value of 0.45.
It will be doubled with a factor entry of 1.8.

Enter factor size ==> 0.7
```

Figure 4. The plot size subroutine.

2. Subroutine LDFILE

The subroutine LDFILE (Figure 5) is displayed on the screen.



```
-----------------------------------------------------
*** This is the routine to LOAD data from a file ***
-----------------------------------------------------

File name should not exceed 8 characters
File extension should not exceed 3 characters



What is the file name (fn) and extension (ext)?

Enter in form fn.ext (e.g. PARAPLAN.INP) ===>  pil.dat
```

Figure 5. The data load subroutine.

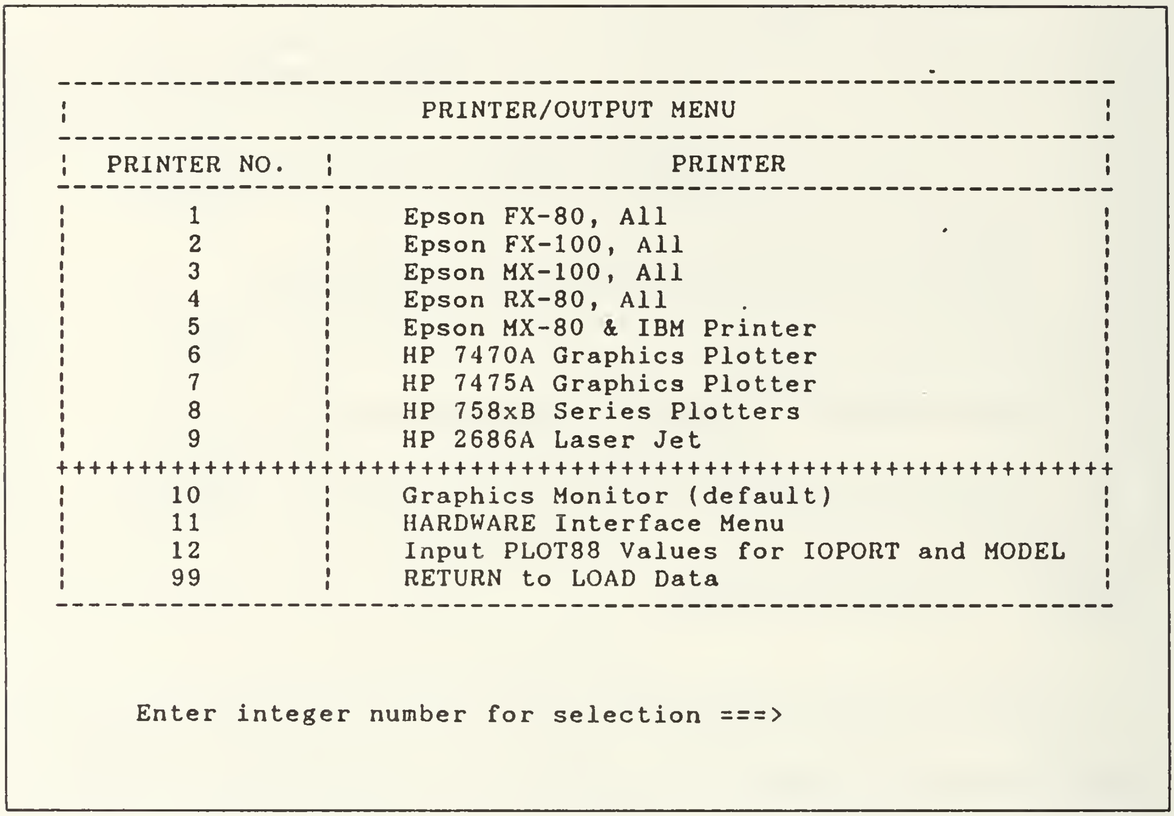
Subroutine LDFILE permits program input of a pre-existing data file. The user is queried as to file name (not to exceed eight characters) and file extension (three characters or less). The code checks for existence of the file within the working subdirectory. If it does not exist, a message is returned stating such and offering the option of entering another file name again. Entry of file extension is optional. In fact, a file can be labeled with as little as one character or number.

We note that the data should be formatted in order to read the data file successfully by using the LDFILE. A data file could be created aligning in the format identified by the subroutine LDFILE source code listed in Appendix B.

3. Subroutine MONPRT

The largest of the subroutines is subroutine MONPRT. A number of smaller subroutines are associated with it. Subroutine MONPRT allows selection of the graphics output device, be it the monitor or a wide variety of printers. Graphics output defaults to the monitor when the program is first executed. Graphics output can always be dumped from the screen to a printer, but there may be times when direct output to a printer is desired. All NPS printers are included in the printer output options. In addition, one option permits the user to directly enter the values for IOPORT and MODEL, as outlined in the PLOT88 manual. [Ref. 3: p. 17]

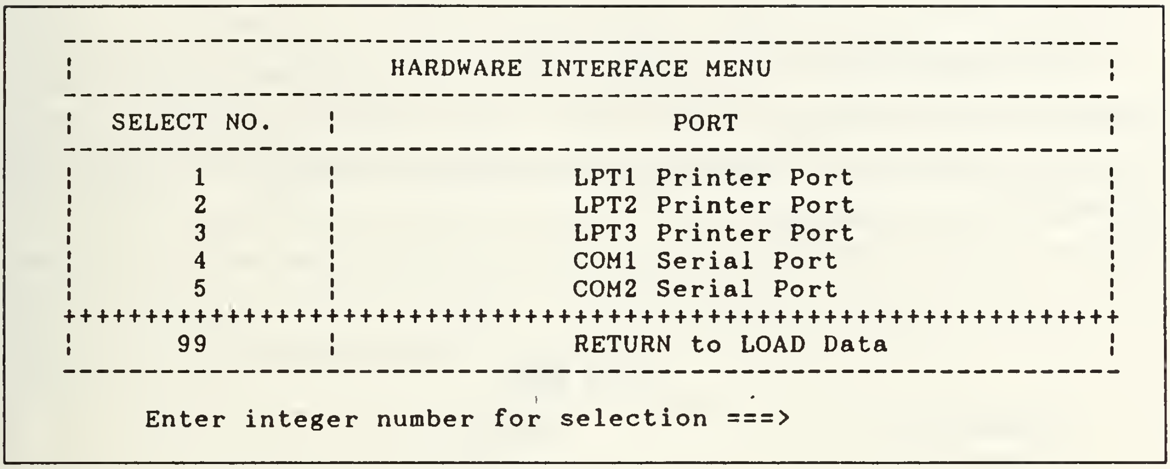
The first menu displayed when subroutine MONPRT is called is the printer/output menu Figure 6.



```
-----------------------------------------------------------------
¦                     PRINTER/OUTPUT MENU                       ¦
-----------------------------------------------------------------
¦  PRINTER NO.  ¦                 PRINTER                       ¦
-----------------------------------------------------------------
¦       1       ¦    Epson FX-80, All                           ¦
¦       2       ¦    Epson FX-100, All                          ¦
¦       3       ¦    Epson MX-100, All                          ¦
¦       4       ¦    Epson RX-80, All                           ¦
¦       5       ¦    Epson MX-80 & IBM Printer                  ¦
¦       6       ¦    HP 7470A Graphics Plotter                  ¦
¦       7       ¦    HP 7475A Graphics Plotter                  ¦
¦       8       ¦    HP 758xB Series Plotters                   ¦
¦       9       ¦    HP 2686A Laser Jet                         ¦
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
¦      10       ¦    Graphics Monitor (default)                 ¦
¦      11       ¦    HARDWARE Interface Menu                    ¦
¦      12       ¦    Input PLOT88 Values for IOPORT and MODEL   ¦
¦      99       ¦    RETURN to LOAD Data                        ¦
-----------------------------------------------------------------


    Enter integer number for selection ===>
```

Figure 6. The printer/output menu.

Selection of a printer in this menu automatically directs graphics output to the most commonly associated output port, either parallel or serial, for that particular device. However, usually there are multiple parallel or serial ports to which a printer can be attached. The default printer output setting may not direct plots to the appropriate port. To provide maximum output flexibility, the printer/output menu allows selection of a specific output port through access to another menu.

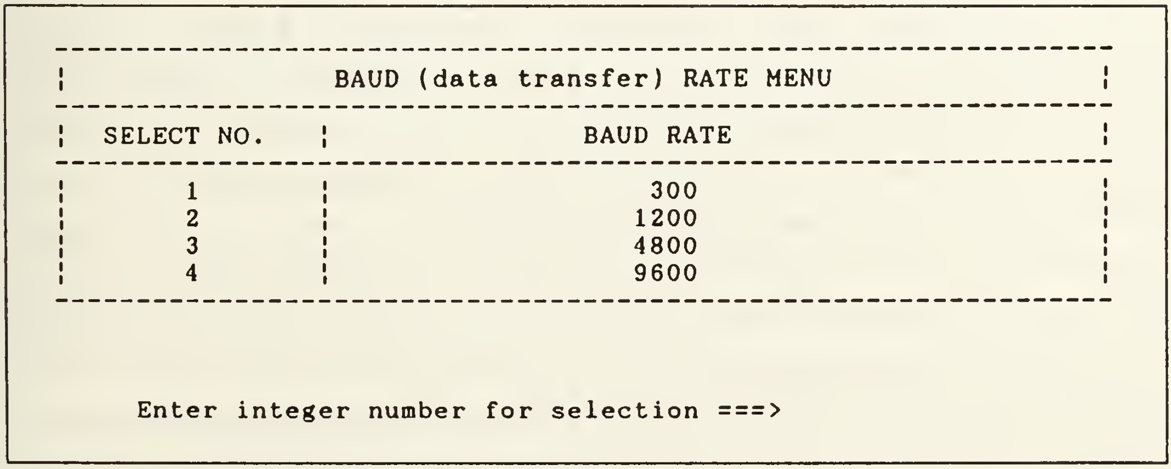


```
-----------------------------------------------------------------
|                    HARDWARE INTERFACE MENU                     |
-----------------------------------------------------------------
| SELECT NO.      |                  PORT                        |
-----------------------------------------------------------------
|       1         |           LPT1 Printer Port                  |
|       2         |           LPT2 Printer Port                  |
|       3         |           LPT3 Printer Port                  |
|       4         |           COM1 Serial Port                   |
|       5         |           COM2 Serial Port                   |
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|      99         |           RETURN to LOAD Data                |
-----------------------------------------------------------------

     Enter integer number for selection ===>
```

Figure 7.   The Hardware Interface Menu.

The output port selection option is called the HARDWARE INTERFACE MENU Figure 7. This menu is part of a subordinate MONPRT subroutine titled PORT. The option allow selection of one of three parallel ports (LPT1 - LPT3) or either of two serial ports (COM1 - COM2).

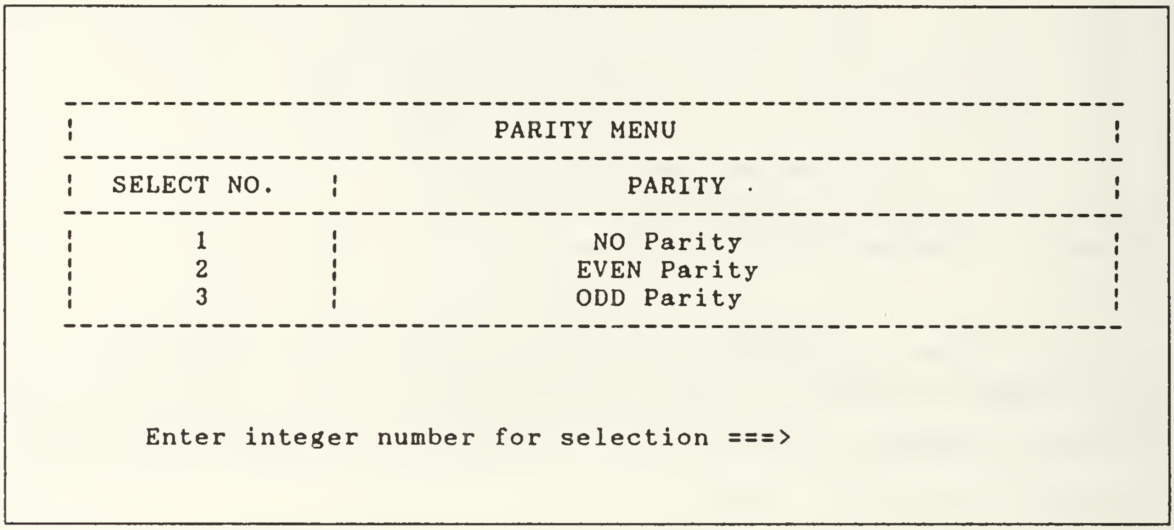
If a serial port is manually selected in the HARDWARE INTERFACE MENU, an associated data transfer (baud) rate must also be assigned. Immediately after selection of serial port options 4 or 5, the BAUD (data transfer) RATE MENU is displayed Figure 8. Transfer of graphics data over a single line is usually time consuming, therefore the highest printer capable transfer rate available (9600 baud) is normally the best selection. A slower transfer rate would only be indicated if improper graphics output is generated.



```
-----------------------------------------------------------------
|                 BAUD (data transfer) RATE MENU                 |
-----------------------------------------------------------------
| SELECT NO.      |              BAUD RATE                       |
-----------------------------------------------------------------
|       1         |                 300                          |
|       2         |                1200                          |
|       3         |                4800                          |
|       4         |                9600                          |
-----------------------------------------------------------------


     Enter integer number for selection ===>
```

Figure 8.   Baud (data transfer) Rate Menu.

Bit by bit transfer of data can also have a check sum (parity) associated with it to provide an interval check of correct data transfer. Following selection of a data transfer rate, the PARITY MENU Figure 9 is displayed. The user has the option of choosing odd,even or no parity.



```
-----------------------------------------------------------------
¦                          PARITY MENU                          ¦
-----------------------------------------------------------------
¦ SELECT NO.     ¦                PARITY                        ¦
-----------------------------------------------------------------
¦       1        ¦               NO Parity                      ¦
¦       2        ¦              EVEN Parity                     ¦
¦       3        ¦              ODD Parity                      ¦
-----------------------------------------------------------------


     Enter integer number for selection ===>
```

Figure 9. Parity Menu.

The PLOT88 Software Library Reference Manual [Ref. 3: p. 18] contains a number of tables which provide common output settings for many more printers than can be selected in the PRINTER/OUTPUT menu.

C. ANSI MODULE

The ANSI module contains two subroutines. Subroutine CLR clears the monitor screen. It is used primarily as a precursor to displaying a menu or positioning queries or informational text. Subroutine CUP positions the cursor immediately after a request for response. This subroutine serves not only to give a pleasing screen display but also provide the user with an immediate reference as to where he is the program and if input is required of the user. Both routines can be found in the PLOT88 Reference Manual [Ref. 3]. The two subroutines which comprise the ANSI module make use of ANSI escape sequences. These sequences are a series of characters that can be used to define functions to MS-DOS. In addition to clearing the screen and positioning, moving, and saving the cursor's position, ANSI escape sequences can control screen graphics and reassign key definitions.

In order to utilize the escape sequences, the microprocessor must first be able to interpret the commands. It does this through a systems file entitled ANSI.SYS. ANSI.SYS must be defined as a device in the CONFIG.SYS file to enable proper interpretation of ANSI escape sequences (e.g., device= \ for \ ansi.sys where \ for \ is the path to ansi.sys). If ANSI.SYS is not resident in the CONFIG.SYS file, the performance index program will not clear the screen or position the cursor when so directed. Instead, the indicated escape code sequence will be printed on the monitor whenever a call is made to CLR or CUP. Although program readability is degraded, the performance index program will still function correctly.

IV. PARAMETER PLANE DESIGN -GRAPHICAL METHOD

A. OVERVIEW

The algebraic solutions have the disadvantage that a fixed value of zeta and omega must first be chosen to compute the alpha and beta terms. In some instances it is possible to modify the remainder polynomial so as to ensure that the specified roots are dominant. However, it is not always possible to guarantee that roots placed at a specified location can be made dominant. Thus, an exhaustive trial-and-error procedure may be required to achieve the best values for the various parameters. To avoid this trial-and-error analytical approach one can employ the graphical solution.

We have two programs available. One permits us to get constant zeta curves and omega curves.The second program gives us a families of constant performance index curves in the parameter plane. We can incorporate the results of two programs by superimposing the plotted performance index cost curves over the parameter plane curves on the same sheet.

To do that, we perform a preliminary procedure. This procedure involves four steps. The first step is to plot the parameter plane curves for the specific problem. This curve permits us to choose ranges of two parameters that are to be used in the analysis and design. The second step is to obtain a data file containing α , β and performance index values by using the D.S.L program. We then copy this data file from the mainframe to a floppy diskette. The third step is to plot the families of performance index curves in the same range of parameter values on the two parameter plane. We then have two kinds of parameter plane curves. The fourth step of the analysis for a given specific problem is to duplicate these curves on the separate transparent sheets and then put them together on the same sheet. Recapitulation of the graphical analysis path is shown in Figure 10.

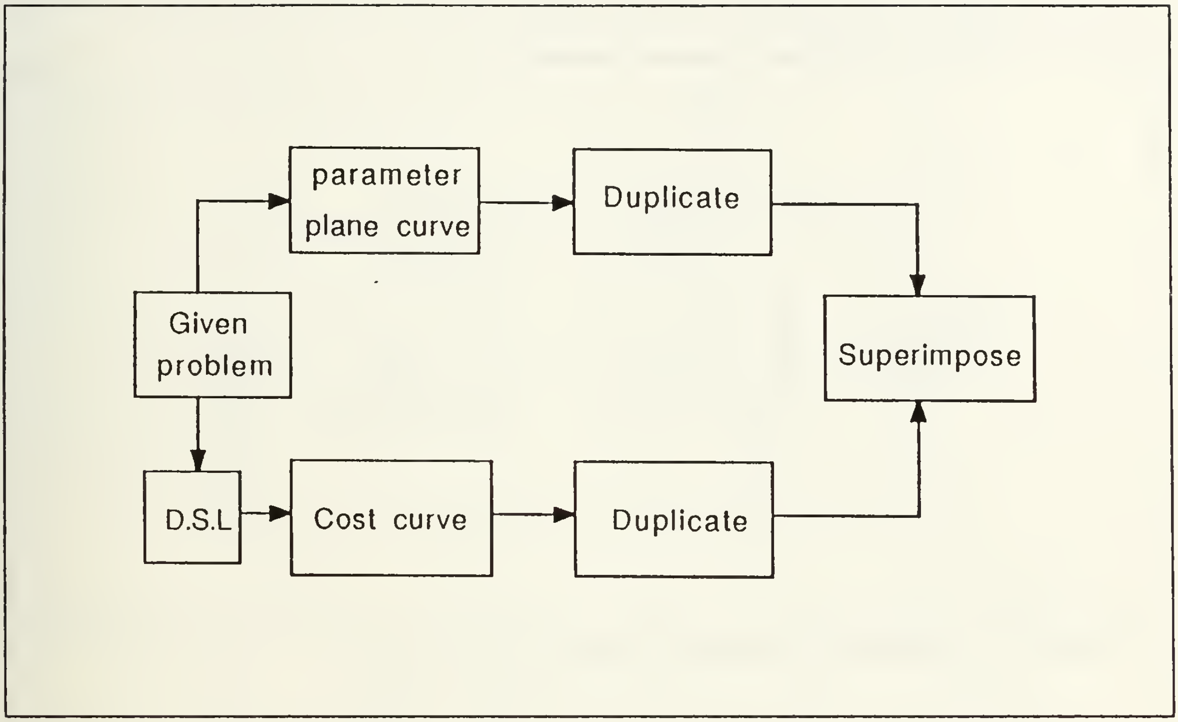


Figure 10. Graphical analysis path.

B. GRAPHICAL SOLUTIONS

In this section, three different kinds of control problems will be demonstrated. One of these is a cascade compensator system. The other one is a velocity and acceleration feedback compensator system. The above two problems will be evaluated with three kinds of performance index constraints, i.e., $\mathrm{PI} = \int E^2\,dt$, $\mathrm{PI} = \int |E|\,t\,dt$ and $\mathrm{PI} = \int |E|\,dt$. The third problem is a steering control for optimized autopilots. The controller parameters are adjusted to minimize the cost function.

$$J = \frac{1}{T}\int_0^T (\lambda\psi_e^2 + \delta^2)\,dt$$

where δ = rudder angle
ψ_e = yaw error
λ = a weighting factor

1. Example 4-1 (cascade compensator)

Figure 11 gives the block diagram of a high gain third order system which is to be damped with a cascade filter. [Ref. 4: p. 319]

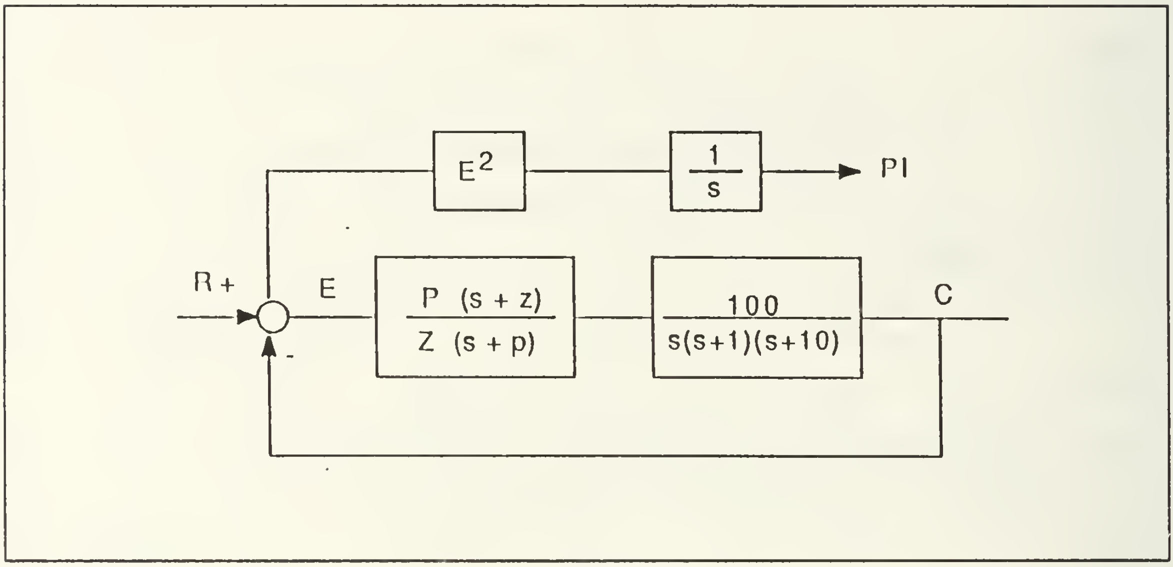


Figure 11. The third order cascade compensated system.

Let

$$\alpha = \frac{P}{Z}$$

$$\beta = P$$

The characteristic equation for this system is then:

$$S^4 + (11 + \beta)S^3 + (10 + 11\beta)S^2 + (10\beta + 100\alpha)S + 100\beta = 0 \tag{4.1}$$

Separating the coefficient terms.

- Constant coefficient: $S^4 + 11S^3 + 10S^2$
- ALPHA coefficient: $100S$
- BETA coefficient: $S^3 + 11S^2 + 10S + 100$

Enter each coefficient array value and desired zeta and omega values into the computer program. Figure 12 provides a look at the combined zeta and omega graph spanning a frequency range from 1.0 to 30, ζ = 1.0 , 0.5 , 0.0 and ω_n = 5.0

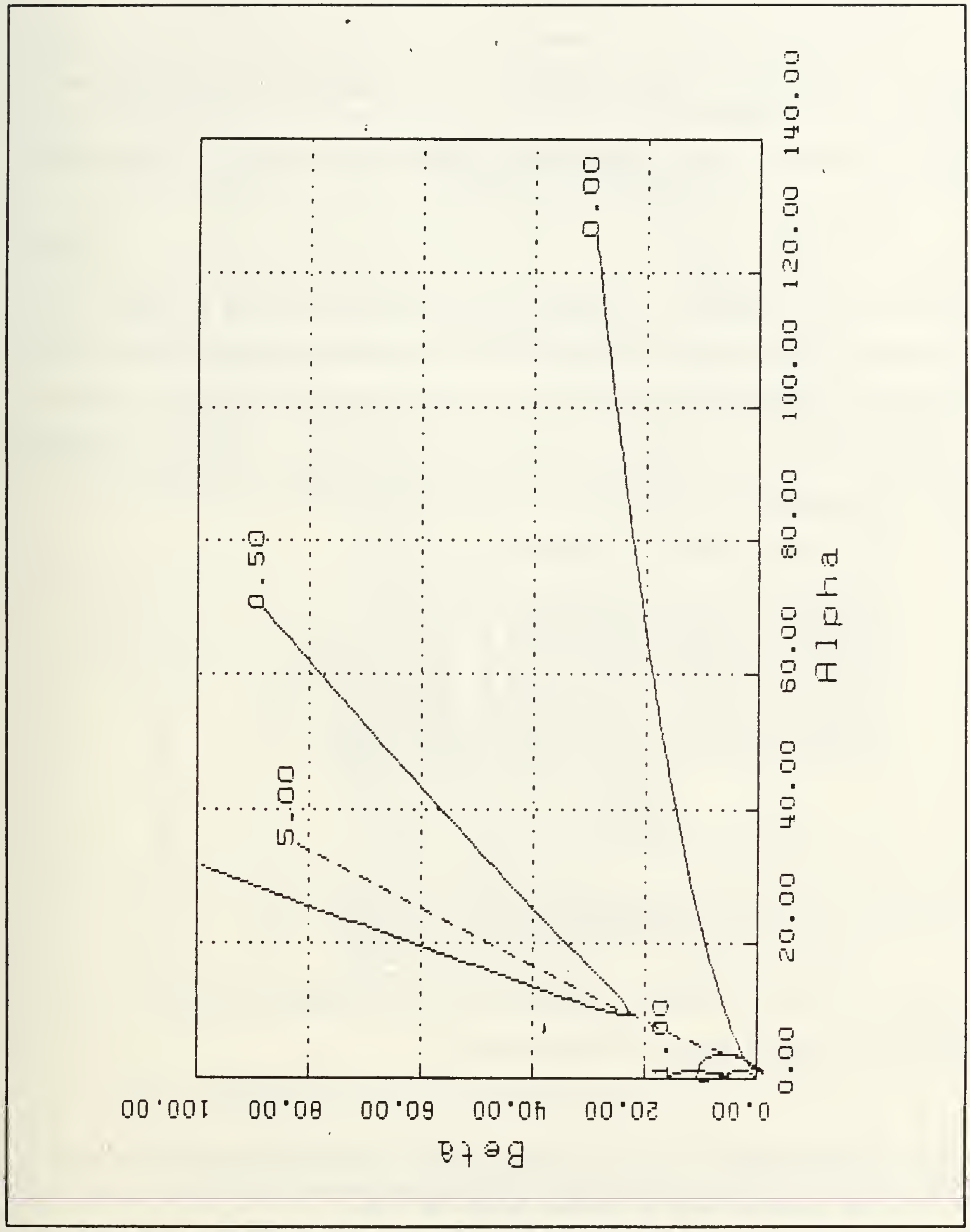


Figure 12. Combine zeta and omega curves for example 4.1

By inspection of the two parameter range, the alpha values vary from 0.0 to 140. and beta values also vary from 0.0 to 100. Now, within these ranges, simulation in D.S.L is used to evaluate the cost at a large number of points on the parameter plane. The D.S.L simulation program and its resulting data file containing α , β and three different criterion cost values are given in Appendix C.

Once we have obtained a data file, we download this file from the mainframe computer to a floppy diskette. To run the performance index program, type in the executable file name, COSTFUN. The IBM microcomputers in the NPS Controls Lab are set up under the file management system 1DIR. To execute the program under the this system, enter the appropriate subdirectory by using the up/down arrow keys to position the cursor adjacent to the subdirectory name, then depress the ENTER key. Use this same procedure to select COSTFUN.EXE and push ENTER to run the program. The preparation step for the plotting desired curves will appear.

Enter proper values after each prompt as follows:

```
1.How many constant PI curves # ===>6
2.Enter PI values ,lwgt,ldig #  1===>0.27,1,2
2.Enter PI values ,lwgt,ldig #  2===>0.2,1,1
2.Enter PI values ,lwgt,ldig #  3===>0.15,1,2
2.Enter PI values ,lwgt,ldig #  4===>0.12,1,2
2.Enter PI values ,lwgt,ldig #  5===>0.11,1,2
2.Enter PI values ,lwgt,ldig #  6===>0.1,1,1
3.Enter number of data file points ===>86
```

Next, adjust the plot size.

```
----------------------------------------------------
*** This is the routine to Adjust the plot size  ***
----------------------------------------------------


Output plots are currently sized to fill the console screen.

Would you like to adjust the plot size?
Enter "y" or "n" ==> y



A default plot factor size of 0.9 is used to fill the console screen.
This plot size will be halved by entering a value of 0.45.
It will be doubled with a factor entry of 1.8.

Enter factor size ==> 0.7
```

The last step is to enter a data file name.

```
-----------------------------------------------------
*** This is the routine to LOAD data from a file ***
-----------------------------------------------------

File name should not exceed 8 characters
File extension should not exceed 3 characters



What is the file name (fn) and extension (ext)?

Enter in form fn.ext (e.g. PARAPLAN.INP) ===>  pil.dat
```

After displaying the graph on the monitor, the program prompts you for a hardcopy of the graph.

```
     WOULD YOU LIKE HARDCOPY OF THESE PLOTS? (Y or N)
y
```

```
-----------------------------------------------------------------
¦                      PRINTER/OUTPUT MENU                       ¦
-----------------------------------------------------------------
¦  PRINTER NO.  ¦                  PRINTER                       ¦
-----------------------------------------------------------------
¦       1       ¦    Epson FX-80, All                            ¦
¦       2       ¦    Epson FX-100, All                           ¦
¦       3       ¦    Epson MX-100, All                           ¦
¦       4       ¦    Epson RX-80, All                            ¦
¦       5       ¦    Epson MX-80 & IBM Printer                   ¦
¦       6       ¦    HP 7470A Graphics Plotter                   ¦
¦       7       ¦    HP 7475A Graphics Plotter                   ¦
¦       8       ¦    HP 758xB Series Plotters                    ¦
¦       9       ¦    HP 2686A Laser Jet                          ¦
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
¦      10       ¦    Graphics Monitor (default)                  ¦
¦      11       ¦    HARDWARE Interface Menu                     ¦
¦      12       ¦    Input PLOT88 Values for IOPORT and MODEL    ¦
¦      99       ¦    RETURN to LOAD Data                         ¦
-----------------------------------------------------------------


     Enter integer number for selection ===>  5
```

Reenter the data file name for the hard copy. Figure 13 is a performance index curve for constraint $\int E^2\,dt$. In like manner, the other two nonlinear criterion of performance index in terms of $\int |E|\, t\, dt$ and $\int |E|\, dt$ can also be plotted by running the same program. Figures 14 and 15 are shown for desired constraint $\int |E|\, t\, dt$ and $\int |E|\, dt$, respectively.

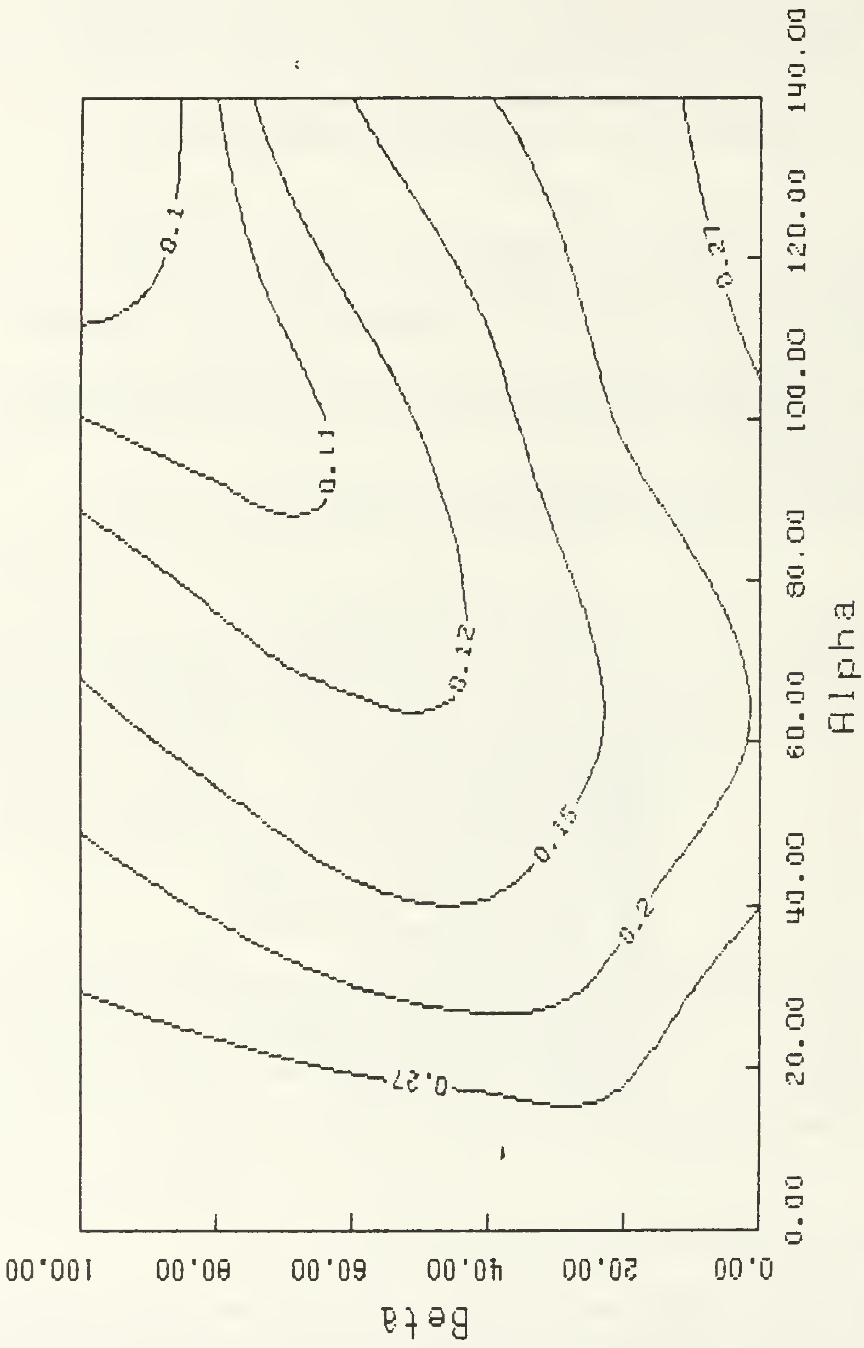


Figure 13. The performance index curves for $\int E^2\,dt$.

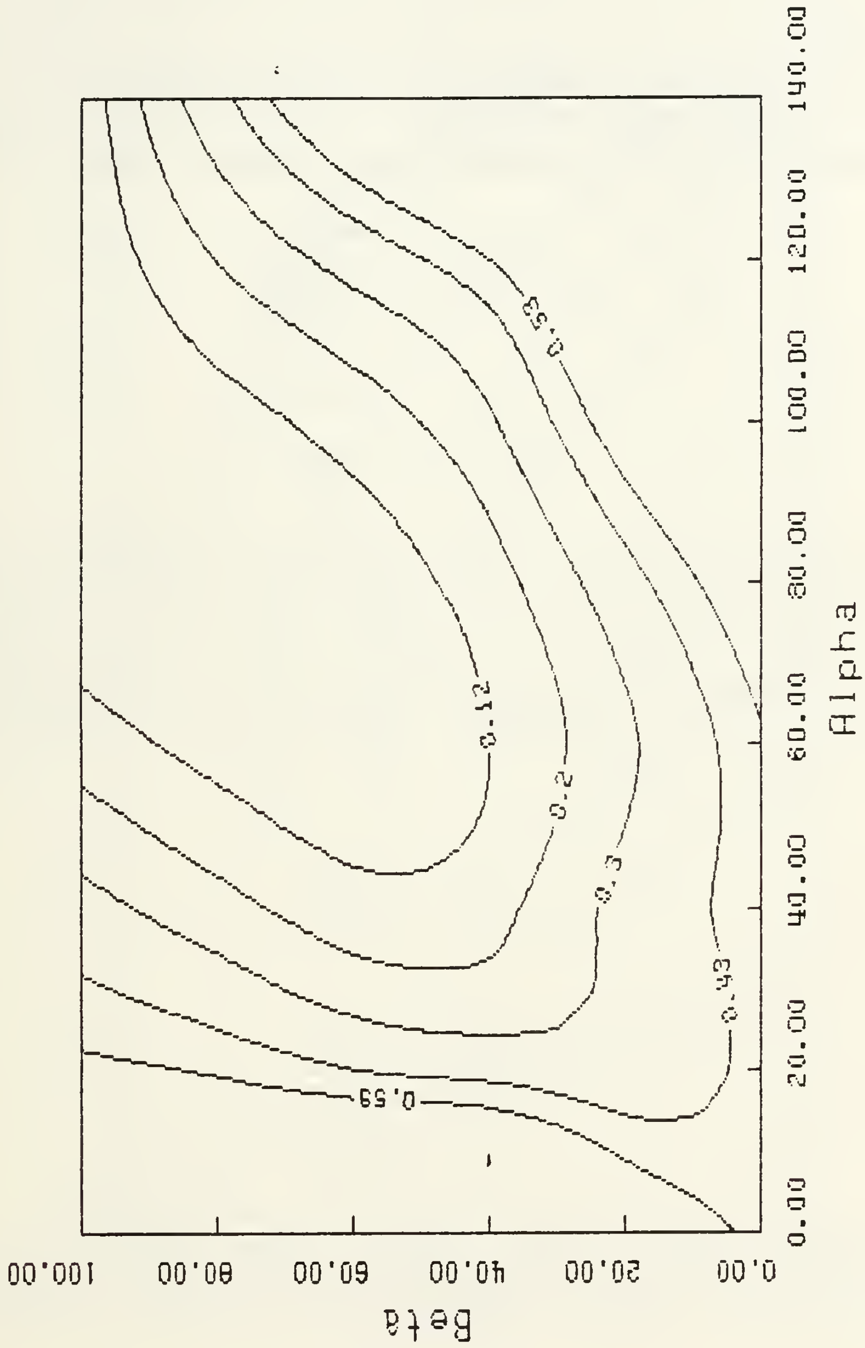


Figure 14. The performance index curves for $\int |E|\, t\, dt$.

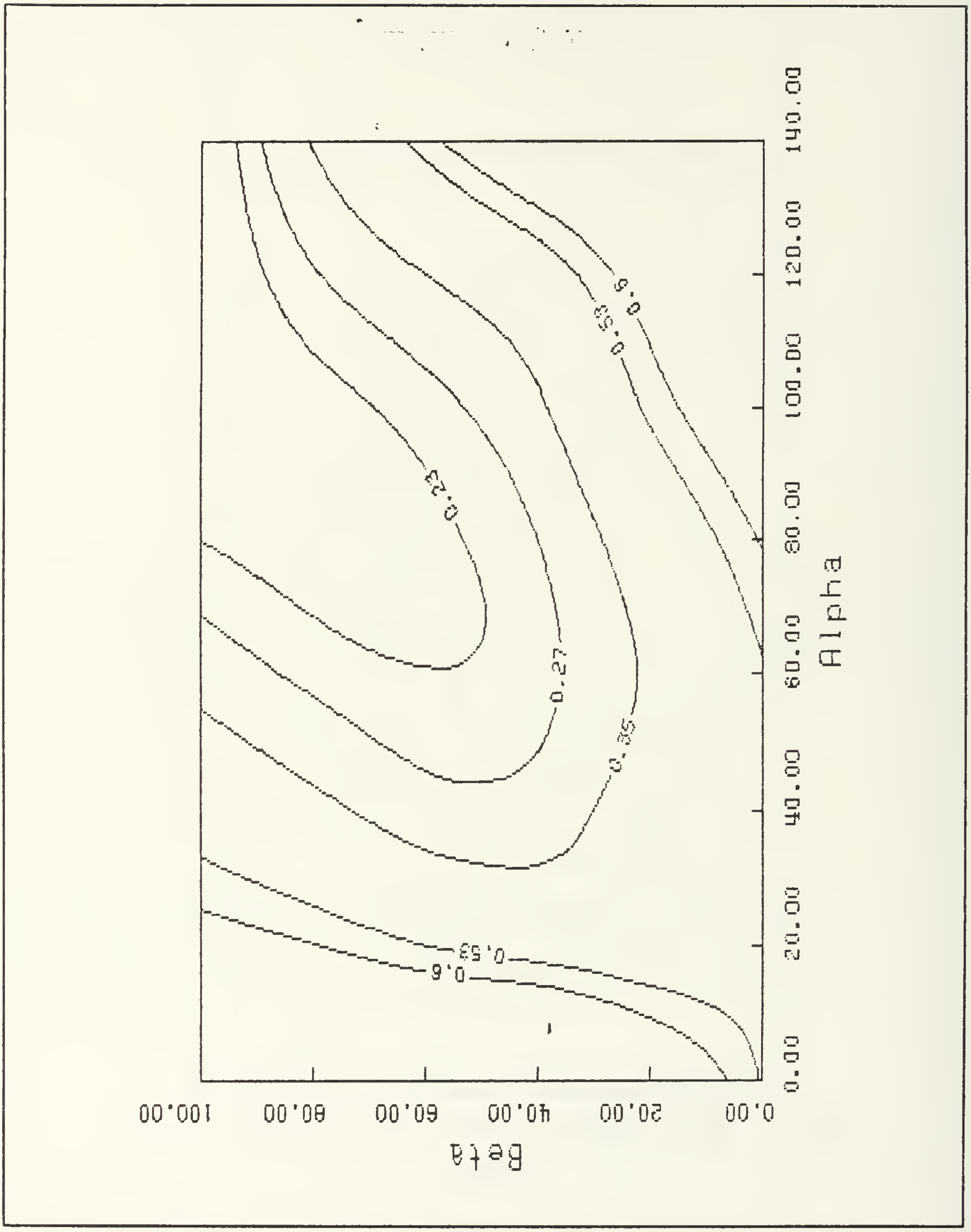


Figure 15. The performance index curves for $\int |E| \, dt$.

Let us consider the two kinds of parameter plane curves, Figure 12 and Figure 13. By superimposing one over the other, the curves can produce the very interesting graph of Figure 16. In like manner, Figure 12 curves superimposed on the other cost contour are shown on Figure 17 and Figure 18. These graphs contain three kinds of constant families curves: ζ , ω_n and cost curves. At this point, from Figure 16, we could select arbitrary operating points. The selected ALPHA/BETA pairs are:

A point: $\alpha = 25, \beta = 80$

B point: $\alpha = 19.5, \beta = 60$

C point: $\alpha = 9.0, \beta = 23$

D point: $\alpha = 100, \beta = 70$

Now, go back to the characteristic equation Eq. (4.1) and substitute in this equation each ALPHA/BETA pair and then, simulate the time response for each one. The tabulated simulation results are shown in Figure 19 and Table 2 on page 32. From the simulation results, we find that operating point D ($\alpha = 25, \beta = 80$) had the fastest setting time, the lowest overshoot and the lowest cost index. The cost associated with this point are:

$$\int E^2\, dt = 0.11$$

$$\int |E|\, t\, dt = 0.12$$

$$\int |E|\, dt = 0.23$$

And step response is given in Figure 19. In this case, a cascade filter was designed with the zero equal to 3.2 (= 80/25) and the pole equal to 80, using $\alpha =$ P/Z and $\beta =$ P.

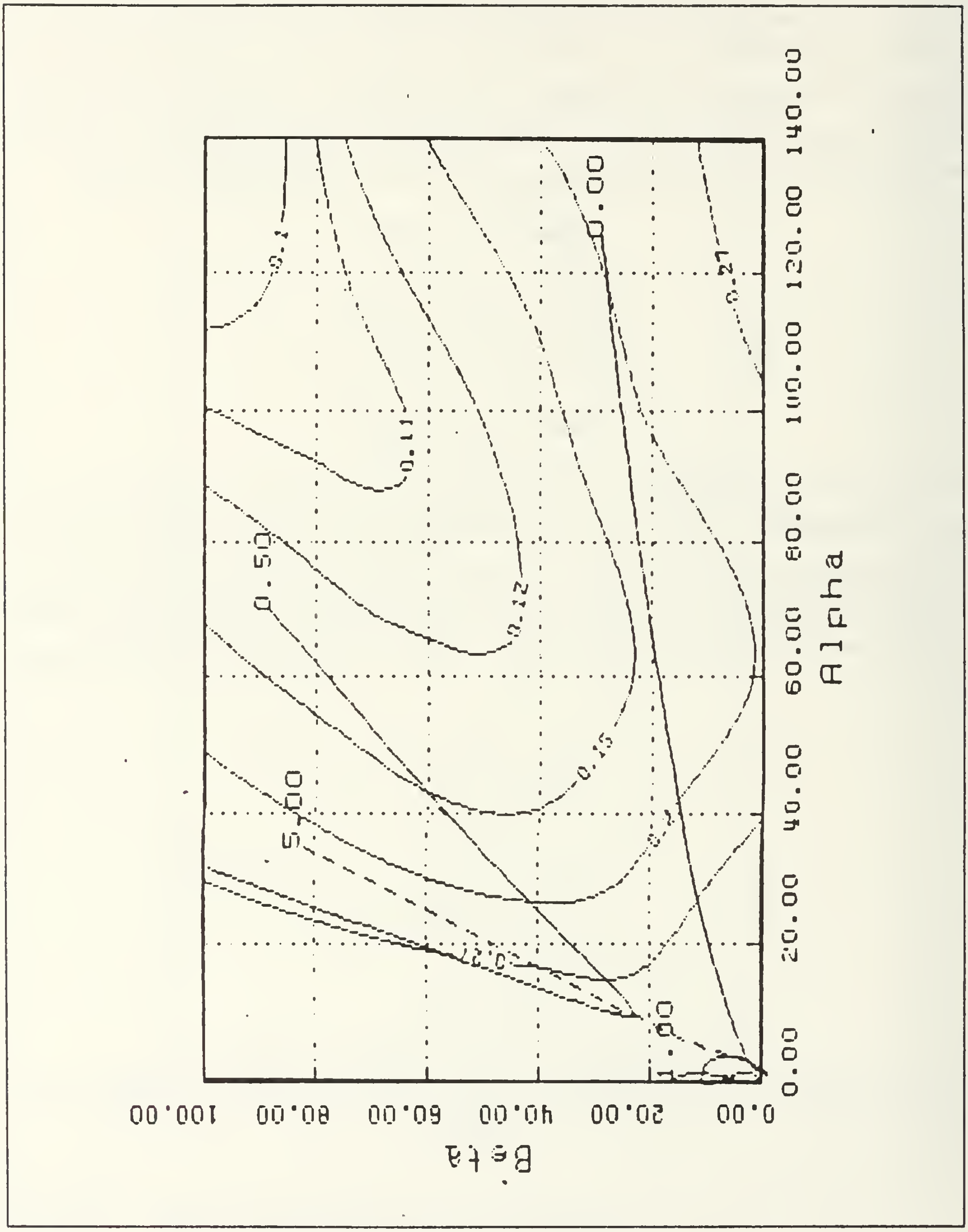


Figure 16. Combine zeta, omega and cost curves for $\int E^2 dt$.

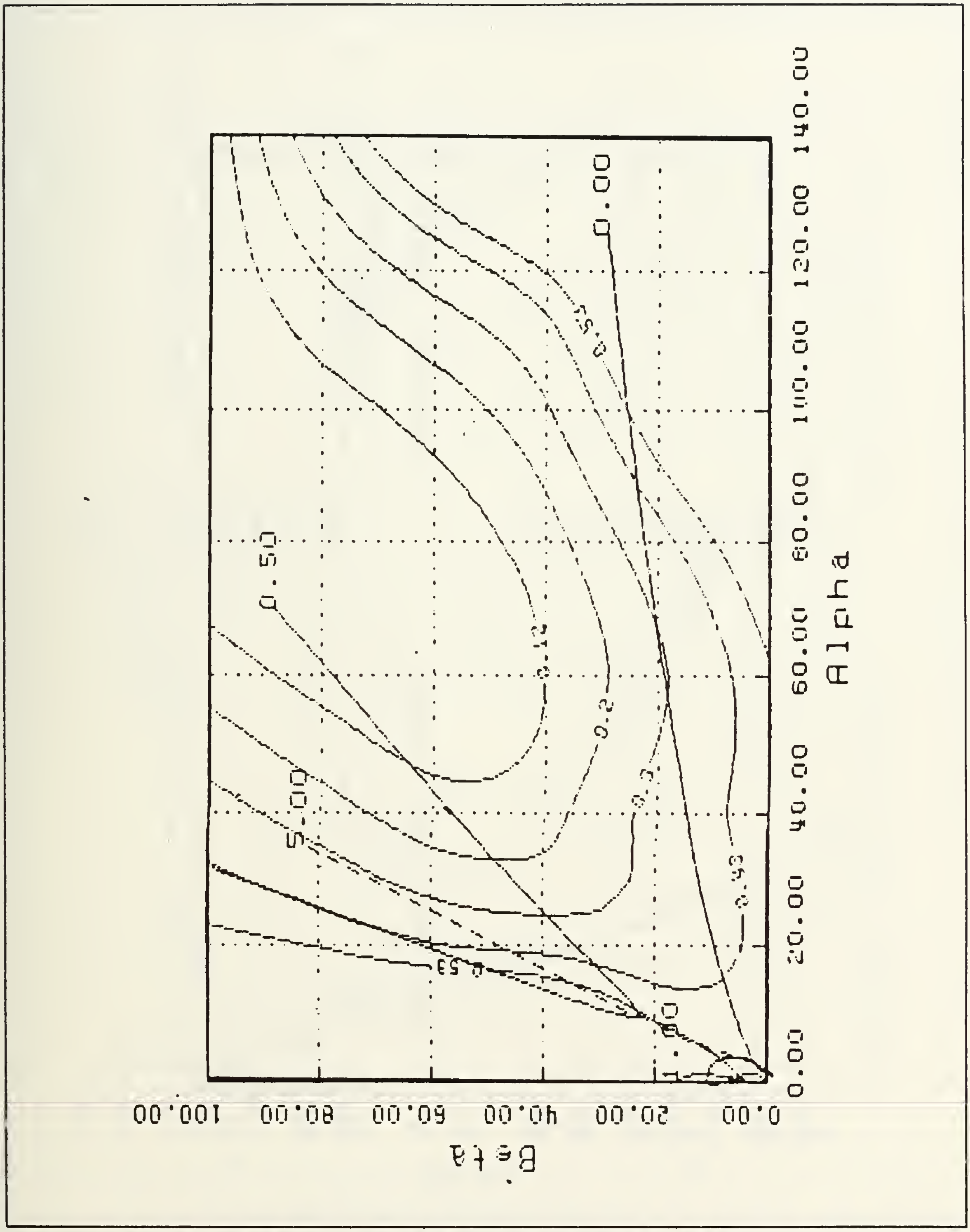


Figure 17. Combine zeta, omega and cost curves for $\int |E|\, t\, dt$.

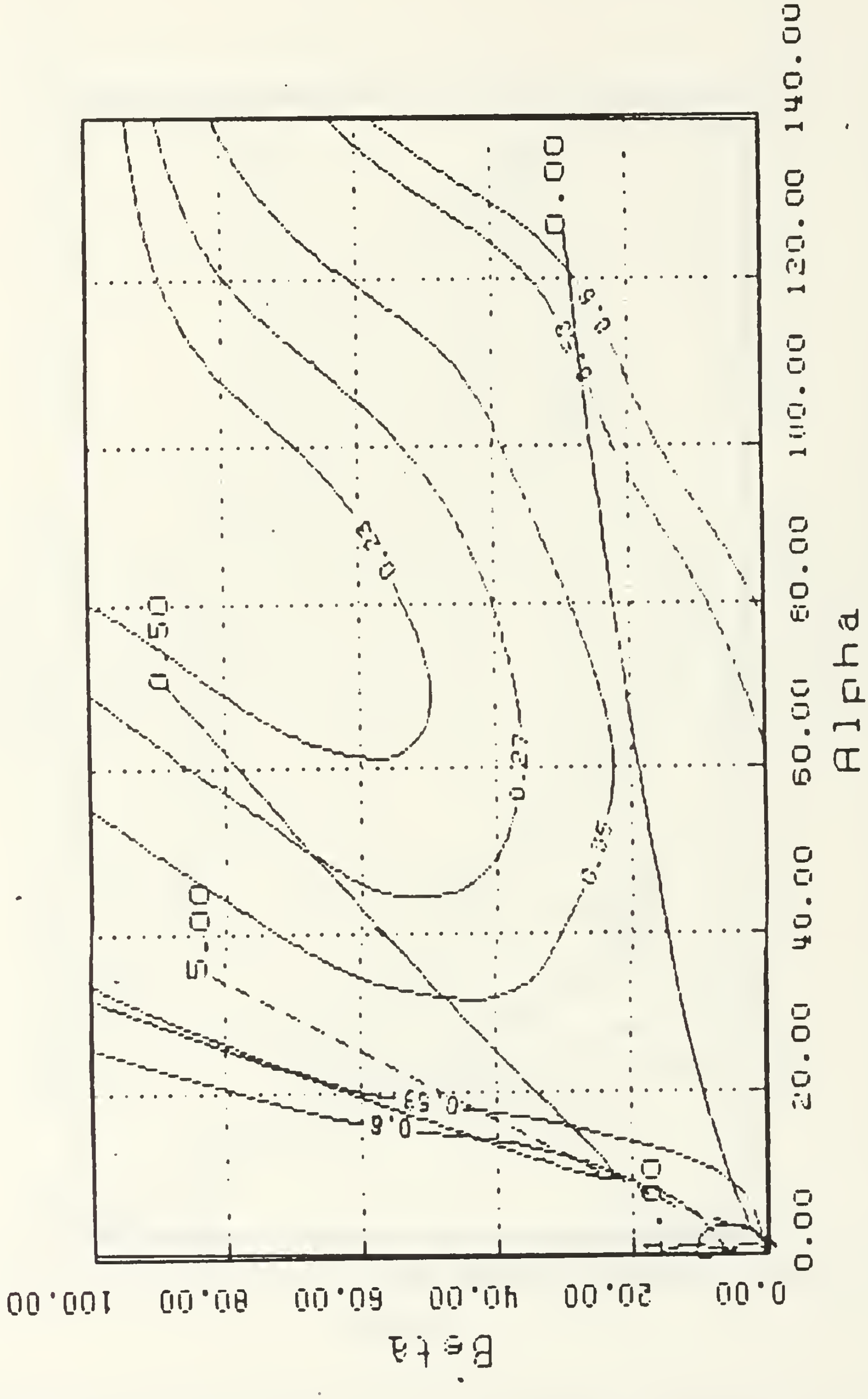


Figure 18. Combine zeta, omega and cost curves for $\int |E|\, dt$.

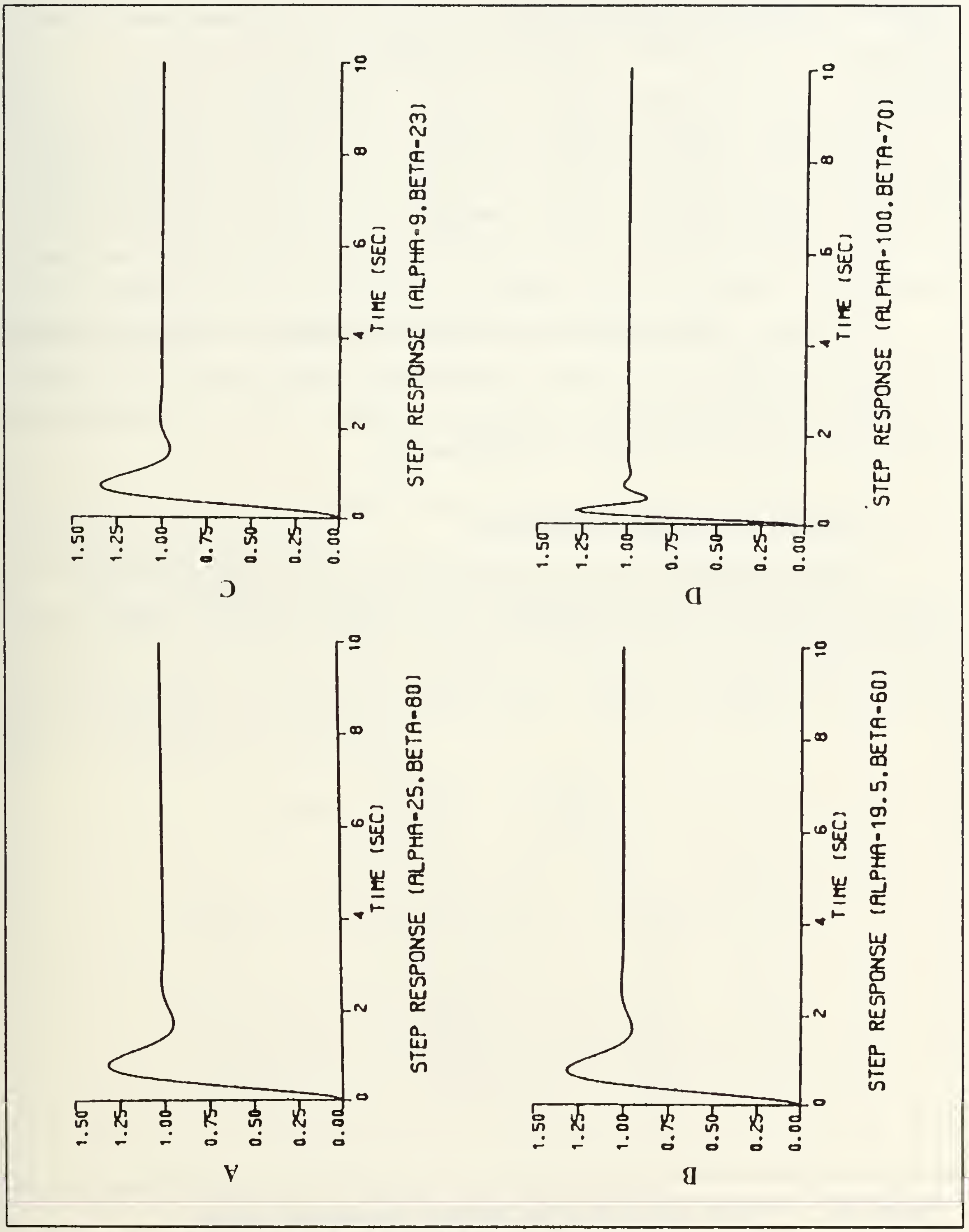


Figure 19. Step response on example 4.1

Table 2. SIMULATION RESULTS.

operating point	parameters		PI values			M_{pt}	t_s
	α	β	$\int E^2\,dt$	$\int \lvert E\rvert\, t\,dt$	$\int \lvert E\rvert\, dt$		
A	25	80	0.27	0.43	0.55	31	2.3
B	19.5	60	0.27	0.43	0.53	31	2.2
C	9	23	0.3	0.53	0.6	34	1.9
D	100	70	0.11	0.12	0.23	29	1.3

The exact values, however, will be chosen to achieve a compromise between transient performance and cost, based on given system specifications. It is of interest to compare the performance index between the constraint $\int E^2\,dt$ and the constraints $\int \lvert E\rvert\, t\,dt$ and $\int \lvert E\rvert\, dt$ at the same operating points.

2. Example 4-2 (feedback compensator)

This example deals with a plant incorporating both velocity (tachometer) and acceleration feedback. The block diagram of such a system is depicted in Figure 20. [Ref. 4: p. 331]

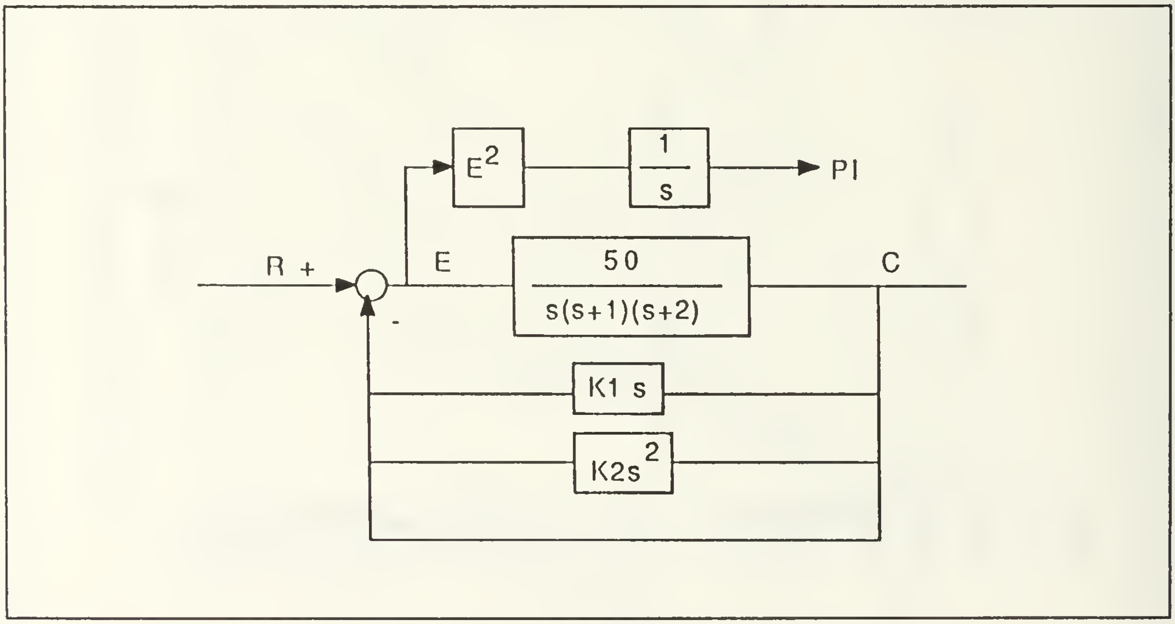


Figure 20. Velocity and acceleration feedback compensated system.

The characteristic equation for this system is:

$$S^3 + (3 + 50K2)S^2 + (2 + 50K1)S + 50 = 0 \tag{4.2}$$

Let

$$\alpha = K1$$

$$\beta = K2$$

Using the same procedure as the previous example, Figure 21 provides the combined zeta and omega graph spanning a frequency range from 1.0 to 30.

$$\zeta = 0.45$$

$$\omega_n = 2.24$$

This graph shows the two parameters range, the alpha values vary from 0.0 to 1.6 and beta values vary from 0.0 to 1.0.

Now, we evaluate the cost at a large number of points on this range by using D.S.L. The D.S.L simulation program and its resultant data file containing α , β and three different cost values are given in Appendix C. Once we have the data file, we can produce the family of performance index curves. These are shown in Figure 22, Figure 23 and Figure 24, respectively.

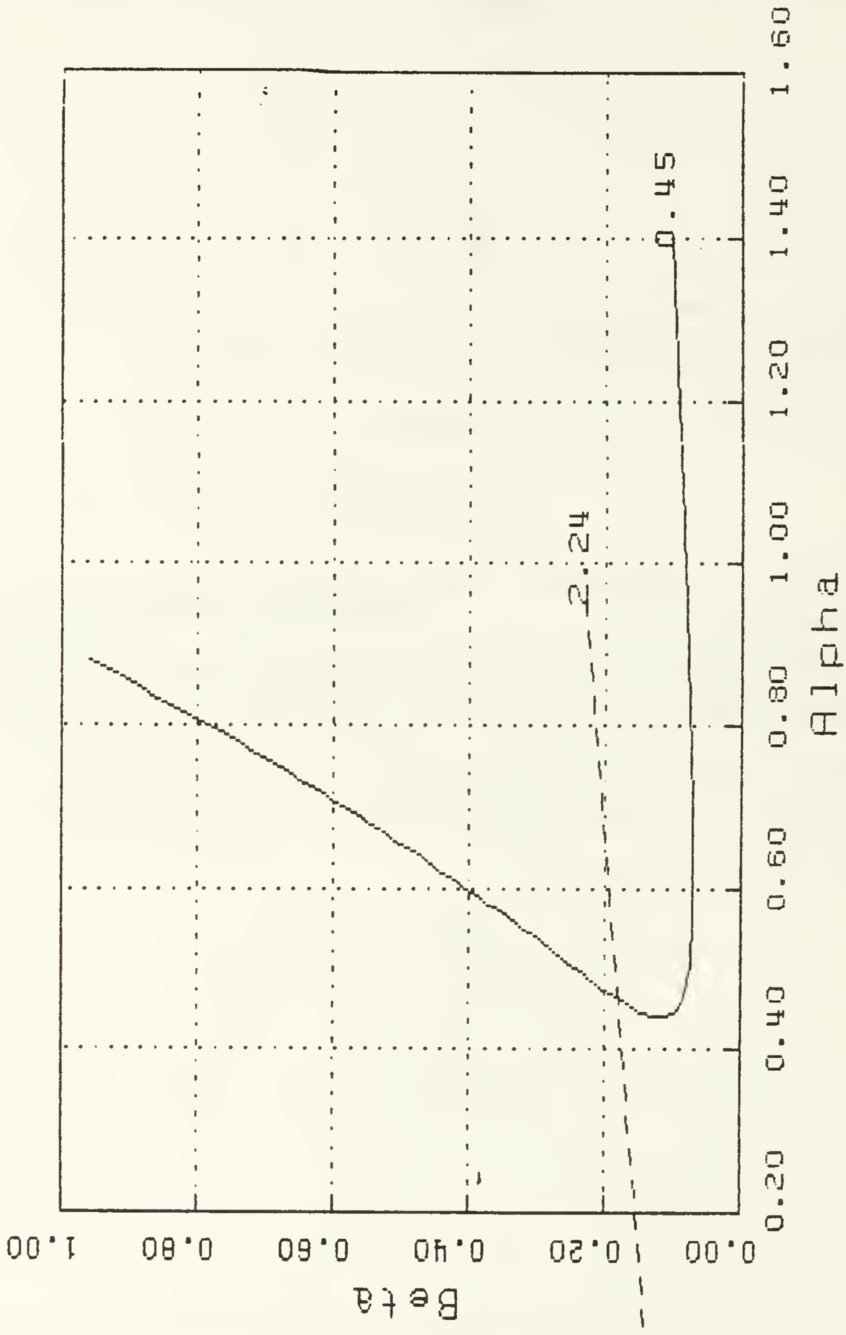


Figure 21. Constant zeta and omega curves for example 4.2

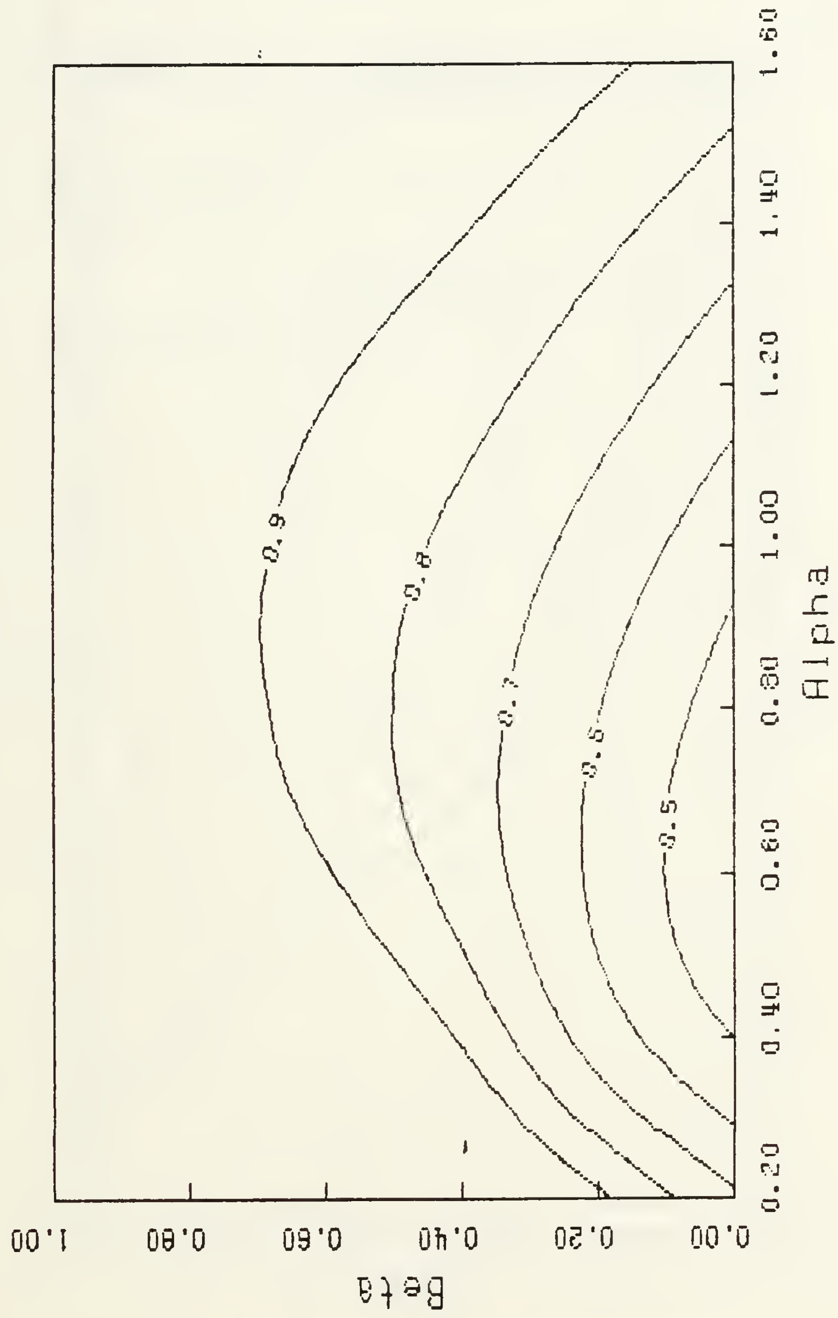


Figure 22. The performance index curves for $\int E^2\,dt$.

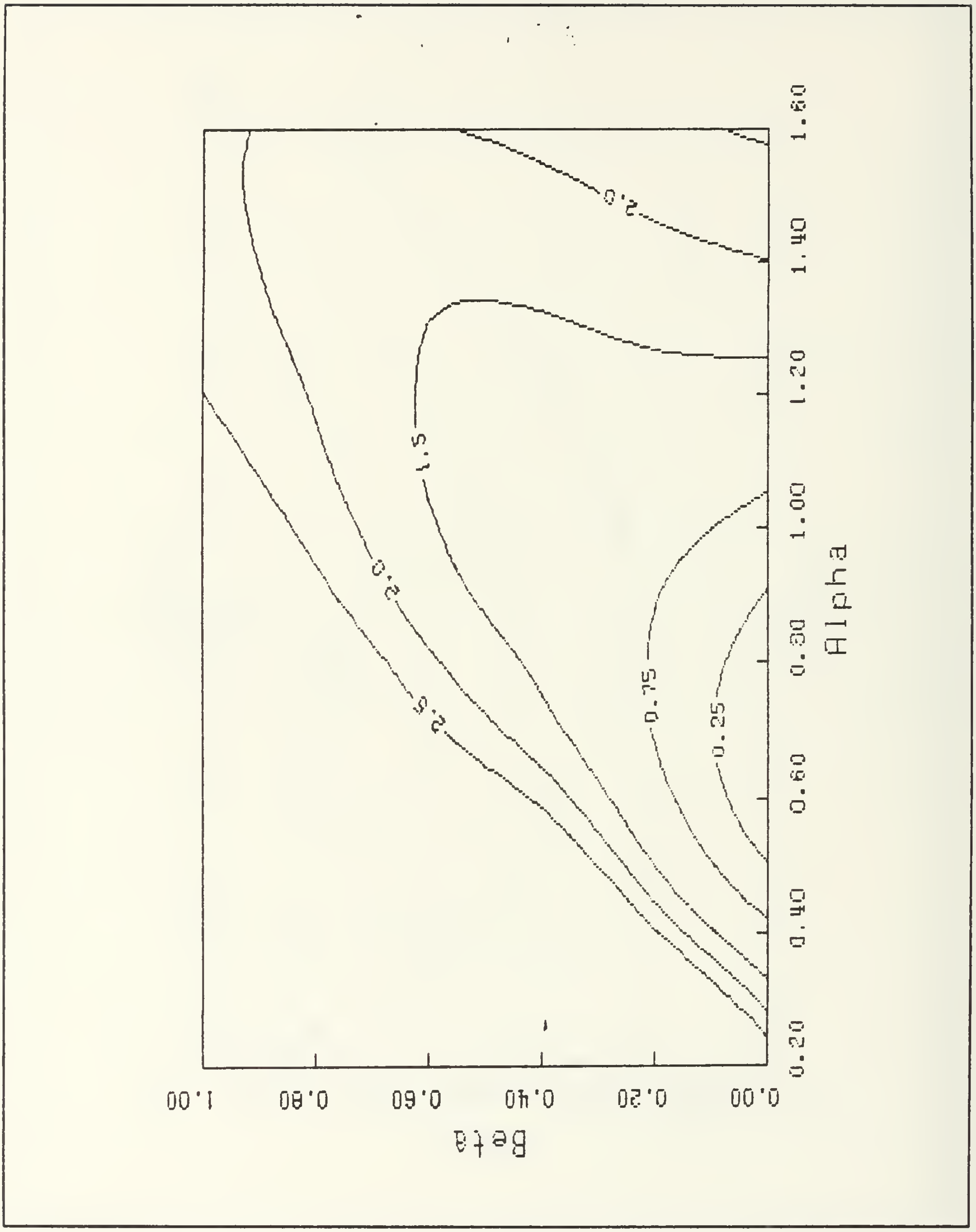


Figure 23. The performance index curves for $\int |E|\, t\, dt$.

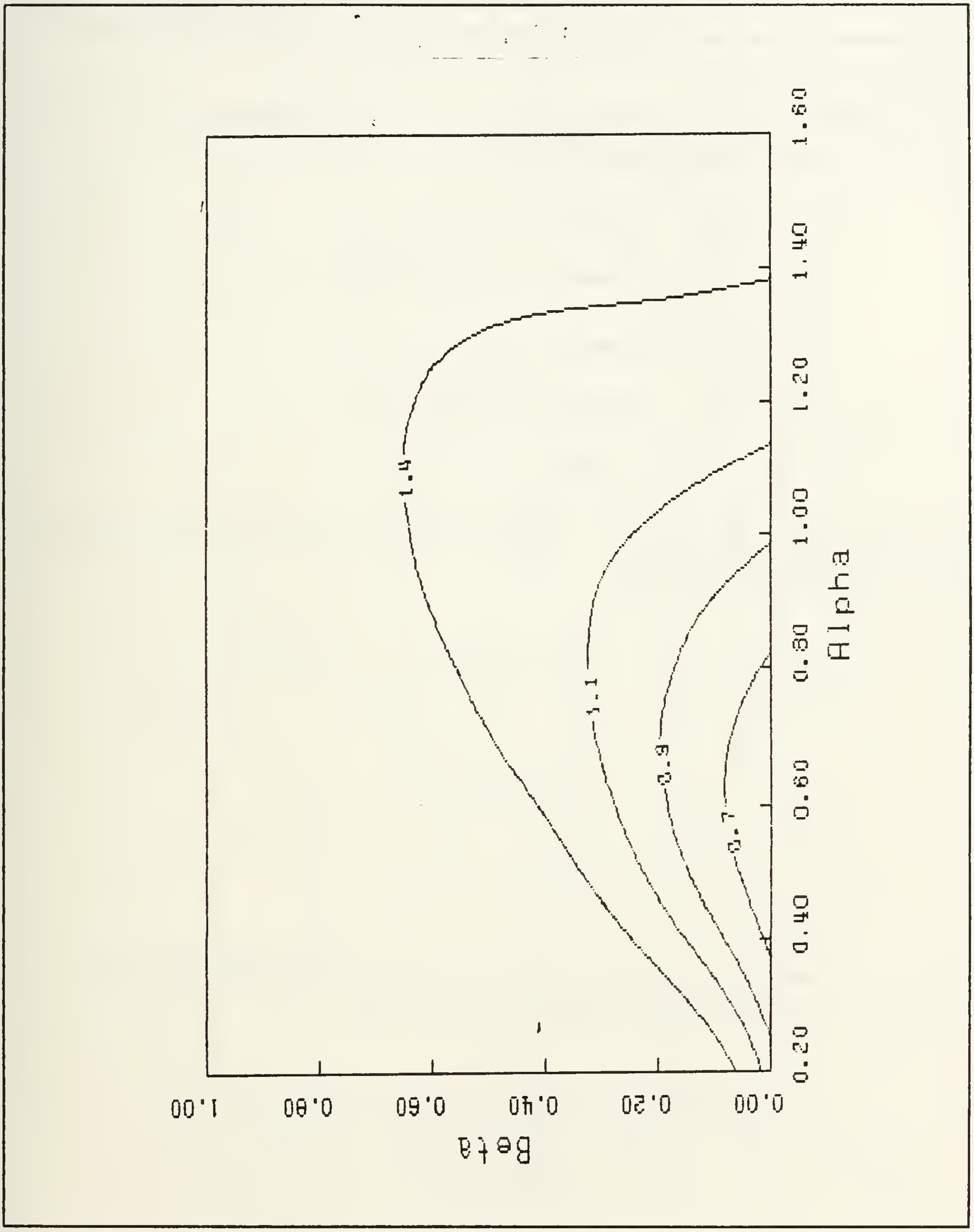


Figure 24. The performance index curves for $\int |E|\, dt$.

Let us consider the parameter plane curves, Figure 25, Figure 26 and Figure 27. These were produced by superimposing Figure 21 over each cost contour curve, Figure 22, Figure 23 and Figure 24.

Figure 25 consists of three families of curves: zeta, omega and cost curves. We could select arbitrary operating points given by:

A point: $\alpha = 0.45, \beta = 0.18$

B point: $\alpha = 0.60, \beta = 0.40$

C point: $\alpha = 0.6, \beta = 0.067$

D point: $\alpha = 0.8, \beta = 0.22$

E point: $\alpha = 0.4, \beta = 0.125$

The operating point C ($\alpha = 0.6, \beta = 0.067$) permits us the lowest cost index. The costs associated with this points are:

$$\int E^2\,dt = 0.47$$

$$\int |E|\,t\,dt = 0.25$$

$$\int |E|\,dt = 0.68$$

However, settling time was too slow, $t_s = 0.6$ sec. In this system, operating point C ($\alpha = 0.6,\ \beta = 0.067$) determines the parameters K1 and K2,

$$K1 = 0.6$$

$$K2 = 0.067$$

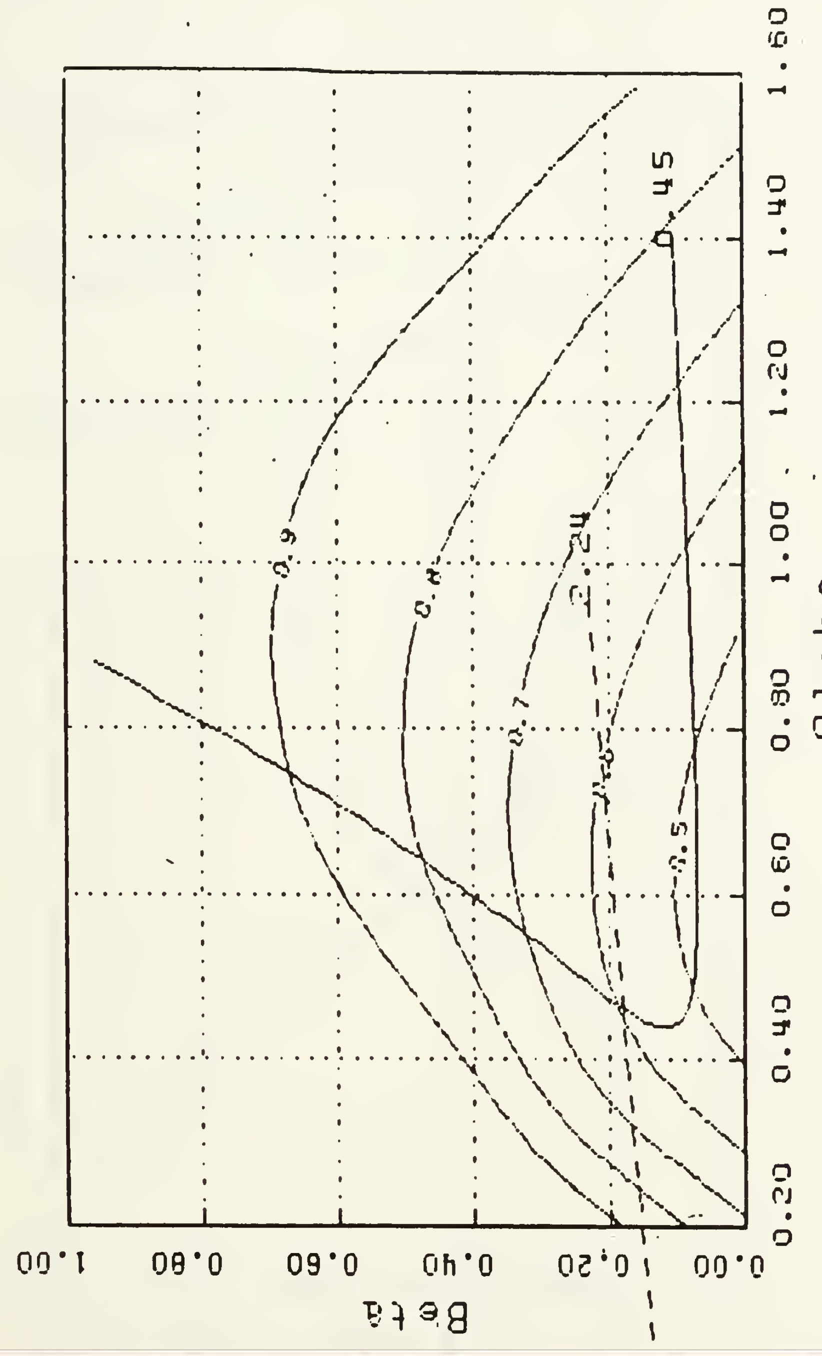


Figure 25. Combine zeta, omega and cost curves for $\int E^2 \, dt$.

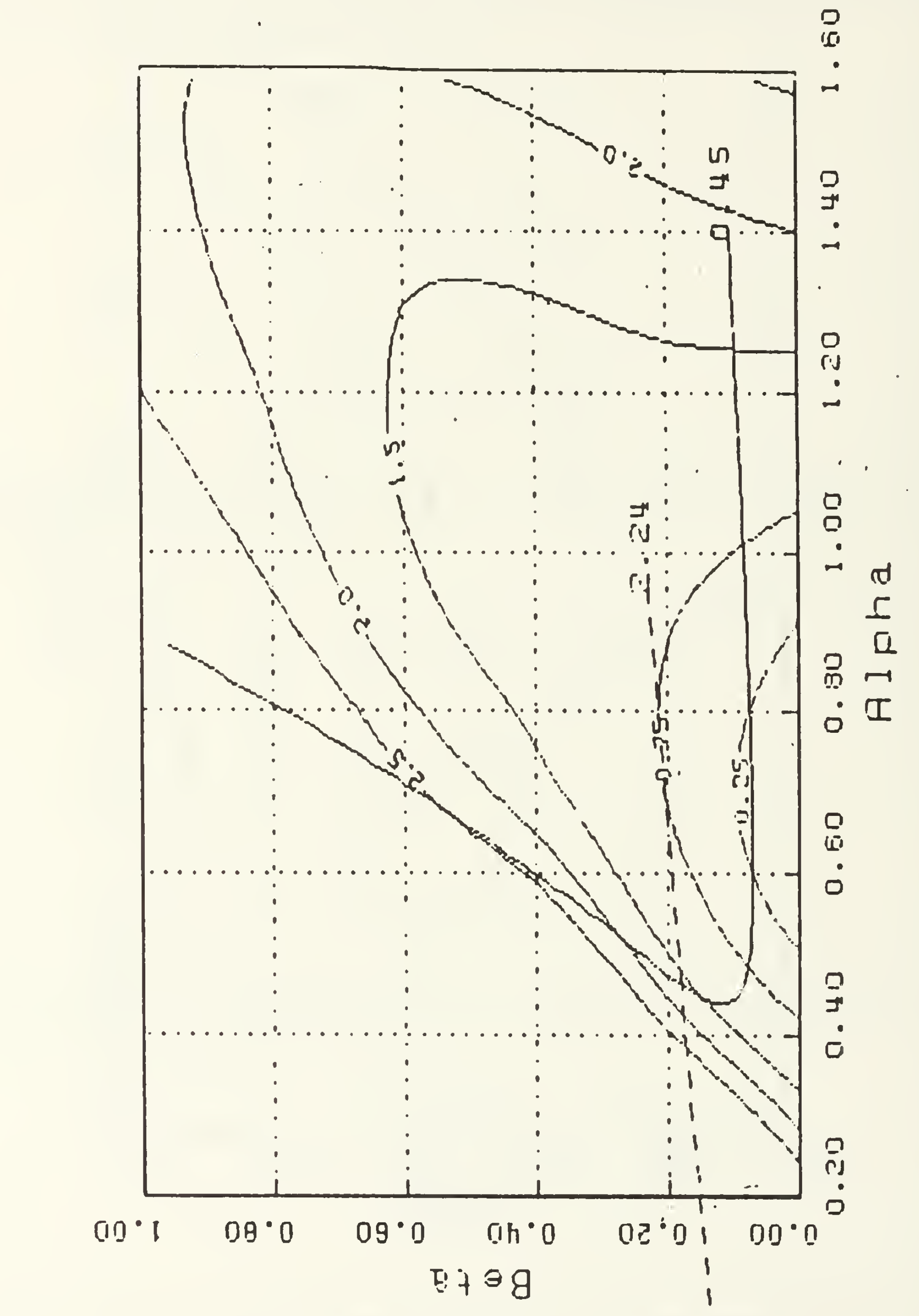


Figure 26. Combine zeta, omega and cost curves for $\int |E|\, t\, dt$.

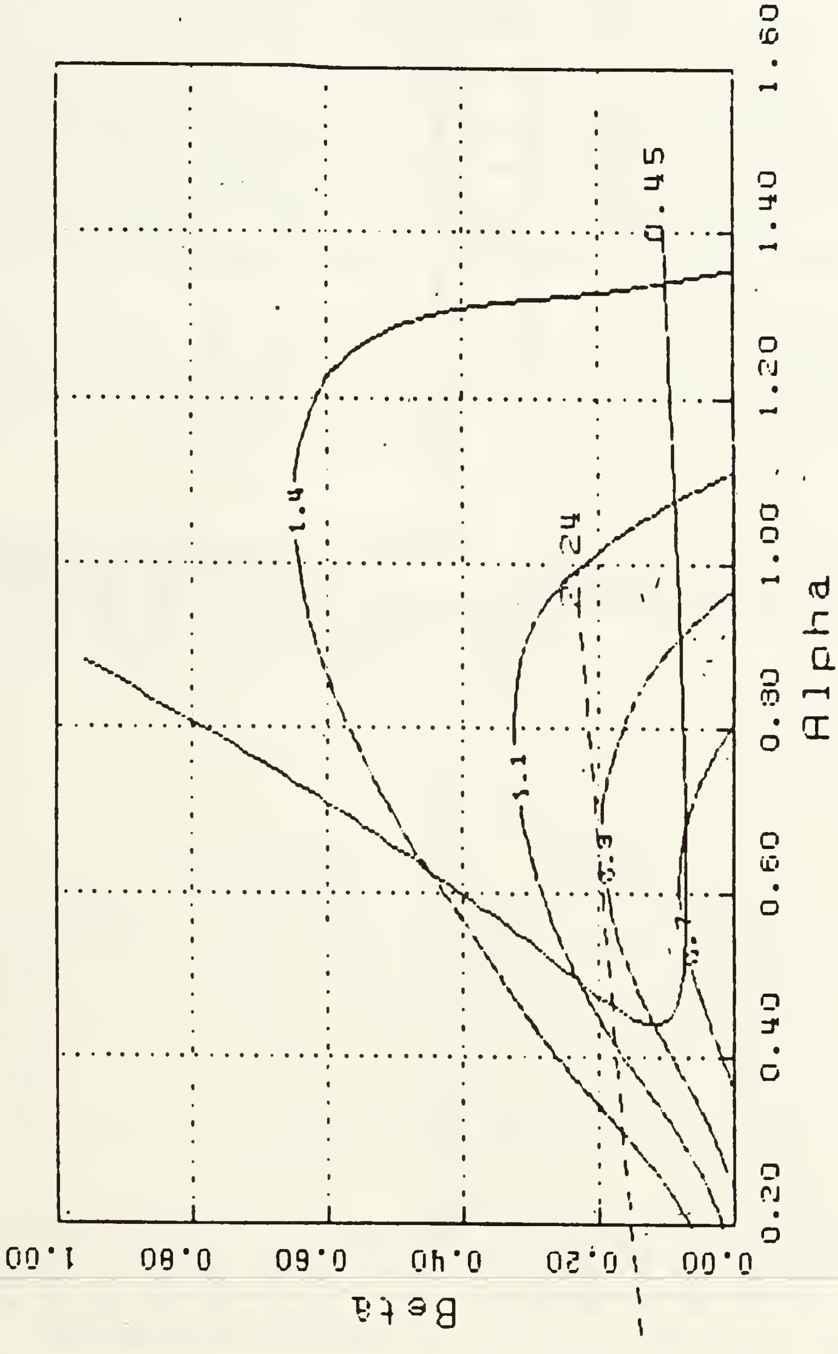


Figure 27. Combine zeta, omega and cost curves for $\int |E|\, dt$.

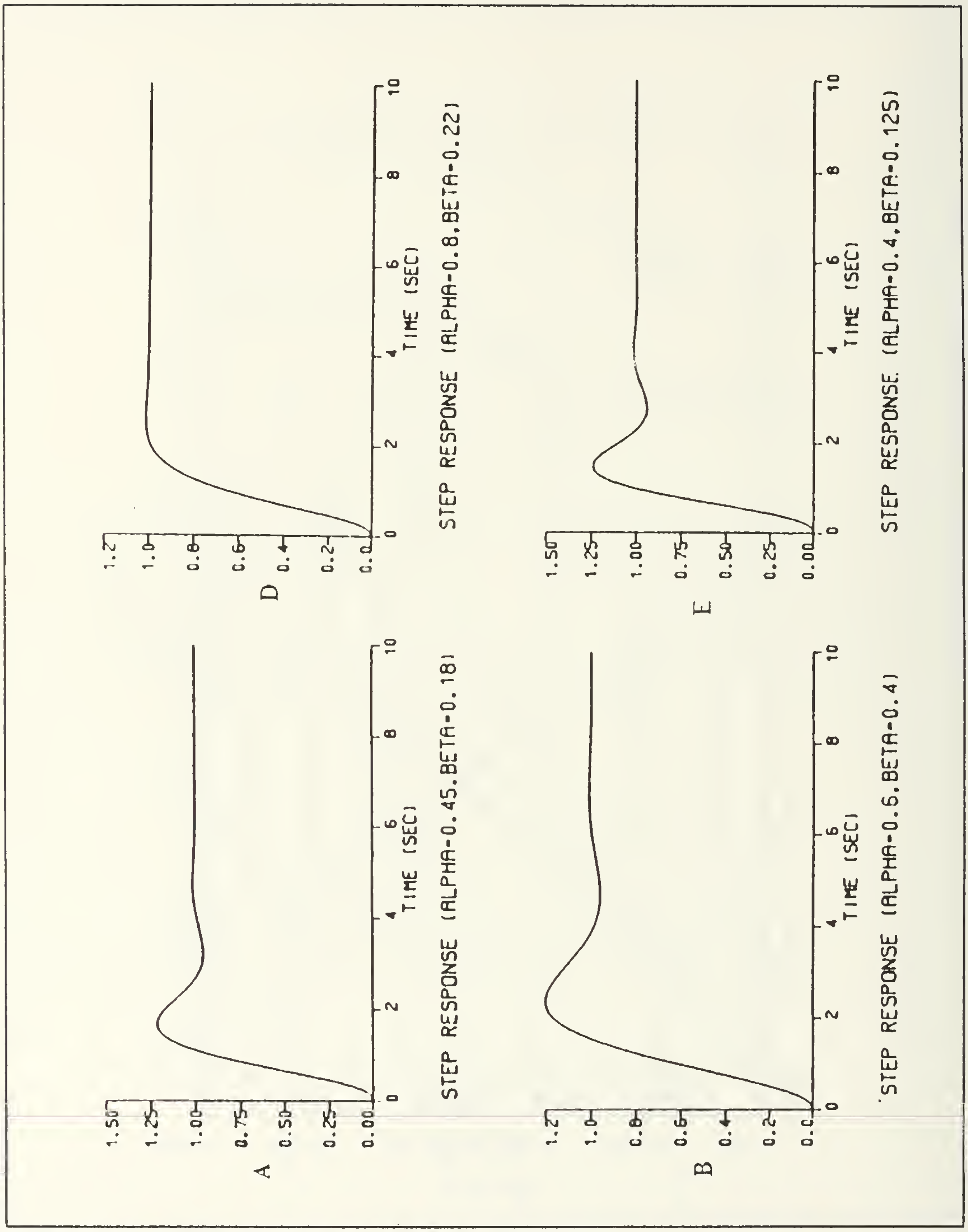


Figure 28. Step response on example 4.2

Table 3. SIMULATION RESULTS.

operating point	parameters		PI values			M_{pt}	t_s
	α	β	$\int E^2\,dt$	$\int \lvert E\rvert\, t\,dt$	$\int \lvert E\rvert\,dt$		
A	0.45	0.18	0.6	1.9	1.1	21	4.1
B	0.6	0.4	0.76	2.5	1.4	20	5.9
C	0.6	0.067	0.47	0.25	0.68	21	6.0
D	0.8	0.22	0.62	0.74	0.95	1	2.1
E	0.4	0.125	0.6	2.0	1.1	23	4.5

Table 3 and Figure 28 show simulation results for operating points A through E. The designer can then choose the operating point (α, β) depending on his desired specifications.

3. Example 4-3 (optimization of controller)

As a final engineering example, consider the steering control for a surface ship. The block diagram of such a system shown in Figure 29.

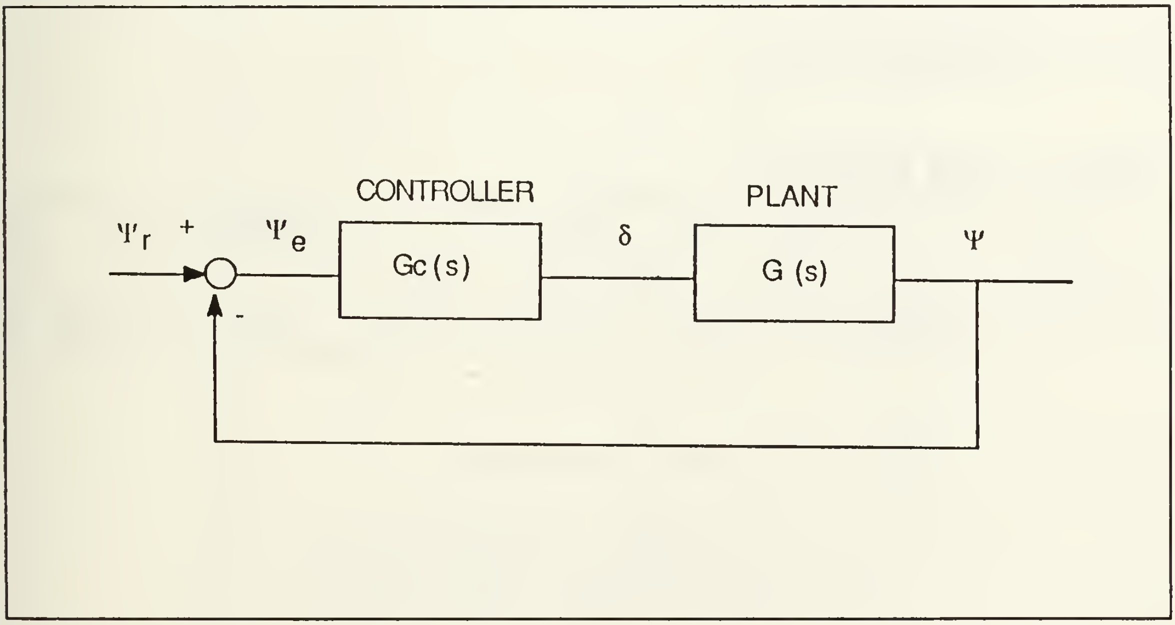


Figure 29. Optimization of controller.

Given that the cost function is defined:

$$\mathbf{J} = \frac{1}{T}\int_0^T (\lambda\psi_e^2 + \delta^2)\, dt$$

where δ = rudder angle

ψ_e = yaw error

λ = a weighting factor

In this study, we used values given by R.E Reid [Ref. 5] for the SL-7 container ship. Reid uses a second order Nomoto model for the SL-7,

$$G(s) = \frac{K}{s(sT + 1)}$$

and also uses a controller

$$G_c(s) = \frac{K1(sT1 + 1)}{(sT2 + 1)}$$

His results are given in Table 4.

Table 4. **REID RESULTS.**

ship speed Knots	plant		weighting factor	controller		
	K	T		K1	T1	T2
16	0.1084	90.36	16.796	0.4556	89.51	10.06

Let us assume T1 and T2 are adjustable free parameters.

$$G \bullet G_c = \frac{0.1084}{s(90.36s + 1)} \bullet \frac{0.4556(sT1 + 1)}{(sT2 + 1)}$$

Let $\alpha = \frac{T1}{T2}$, $\beta = \frac{1}{T2}$

The characteristic equation for this system is:

$$90.36S^3 + (1 + 90.36\beta)S^2 + (\beta + 0.049\alpha)S + 0.049\beta = 0 \tag{4.3}$$

The parameter plane curves examined within the frequency range from 0.01 to 0.1 $\zeta = 0.5, 1.0$ window size; $0 \le \alpha \le 35., 0 \le \beta \le 1$ Using the parameter plane program, we get the praph of Figure 30.

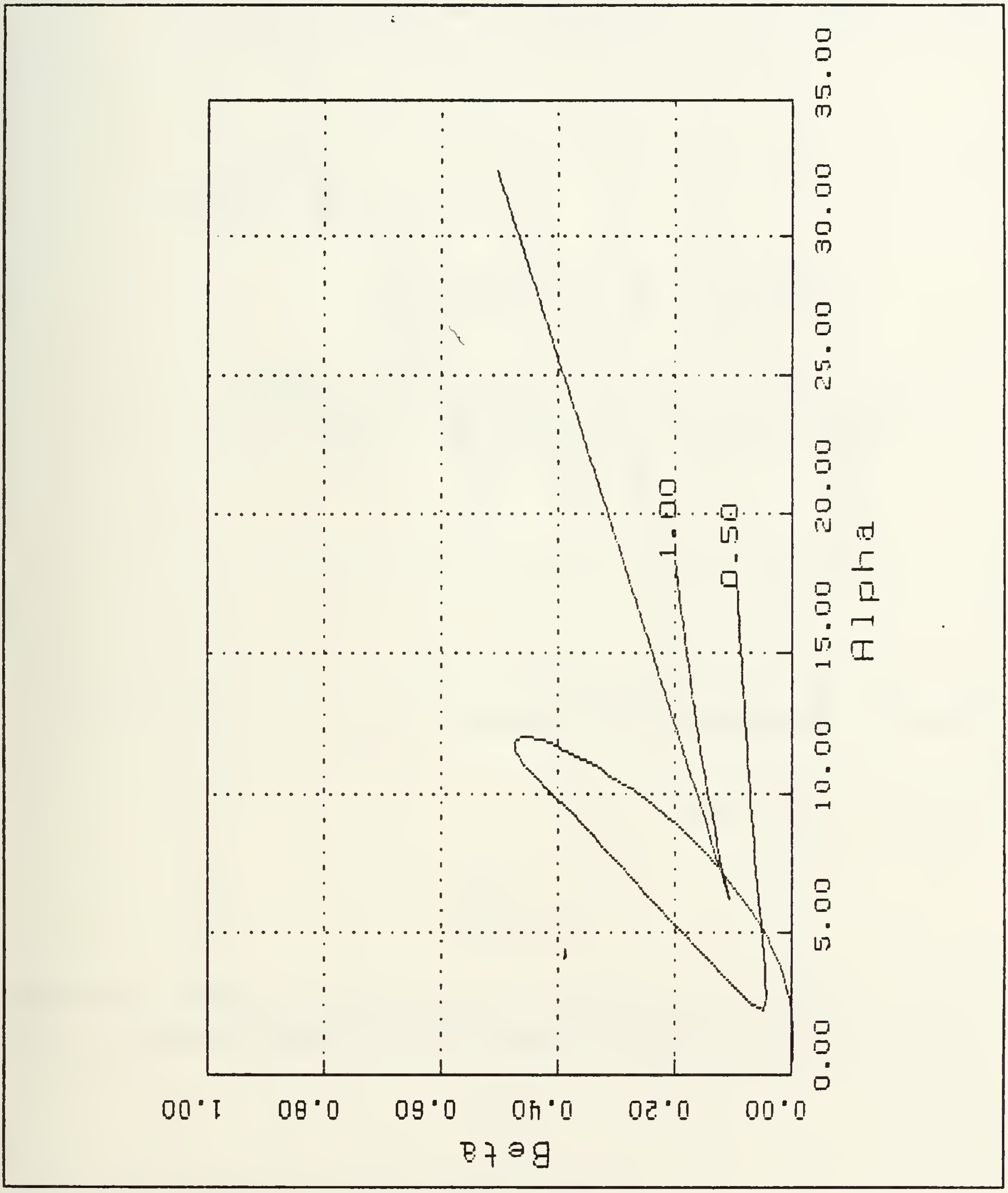


Figure 30. Constant zeta curves for example 4.3

In order to evaluate a family of cost curves, we have to obtain a data file on the same range of parameters, as Figure 30 by simulation. The D.S.L simulation program can be implemented with ease by rearranging the block diagram in Figure 31.

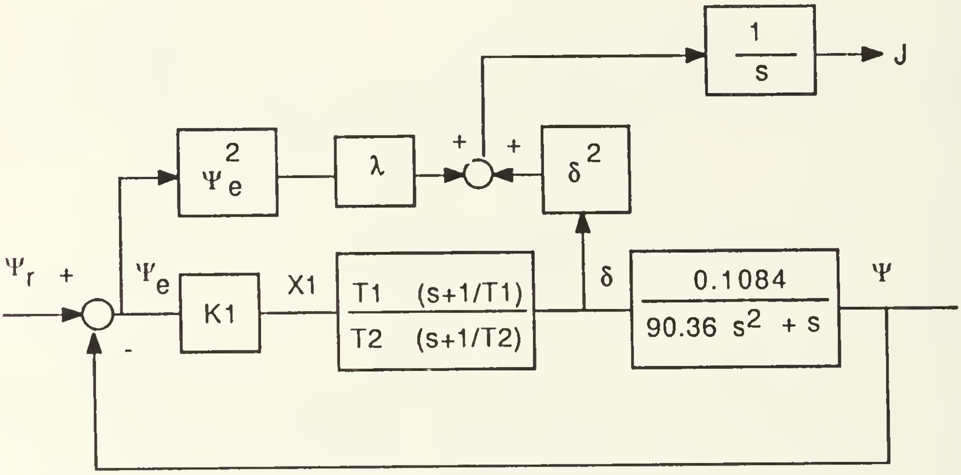


Figure 31. Rearrangement of block diagram.

The controller block becomes

$$\alpha \bullet \frac{(s + \frac{\beta}{\alpha})}{(s + \beta)}$$

The D.S.L simulation program and data file for this example are given in APPENDIX C. Applying the previously discussed procedures for producing constant cost curves provides the graph of Figure 32.

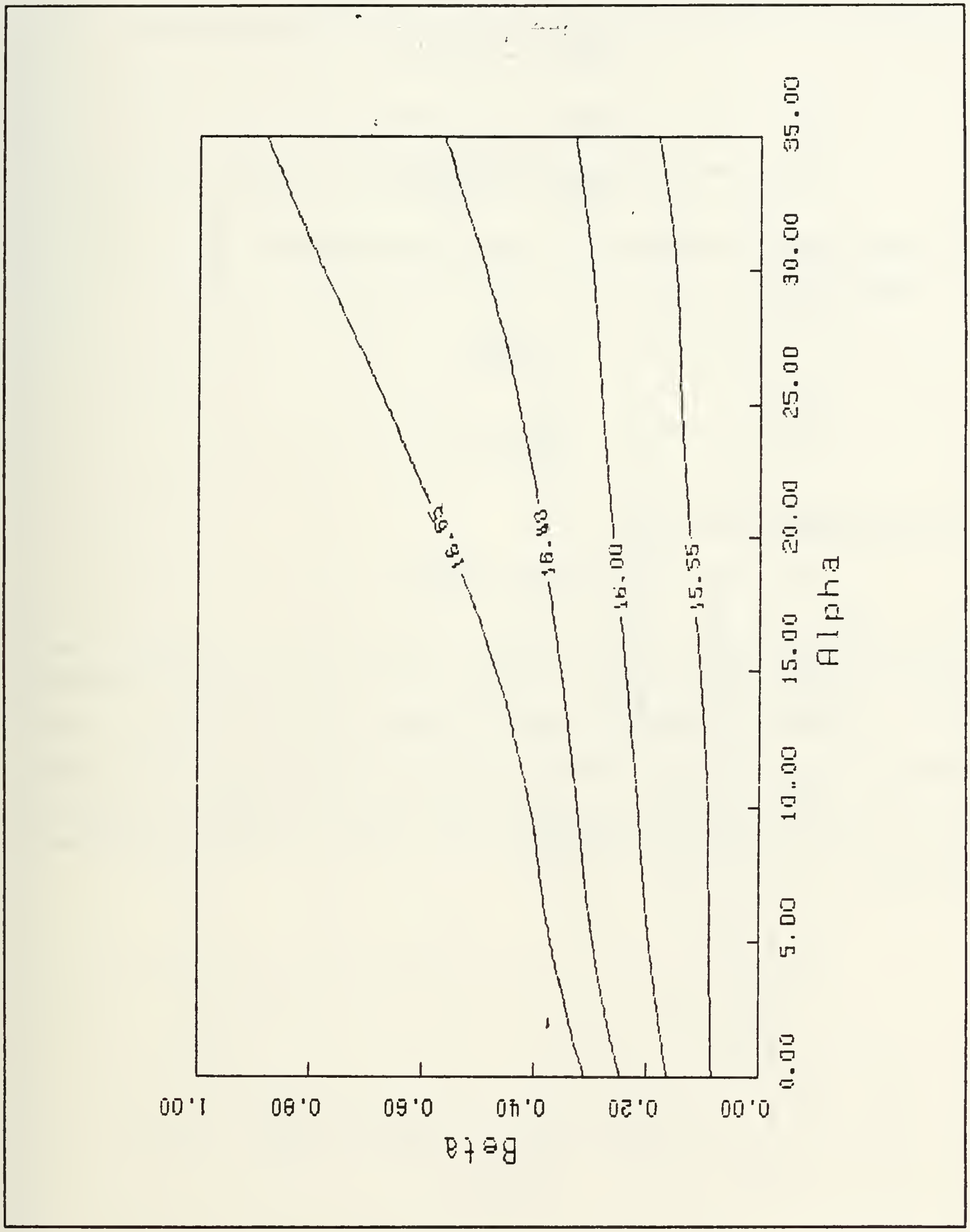


Figure 32. The cost curves for $\frac{1}{T}\int_0^T(\lambda\psi_\varepsilon^2 + \delta^2)\,dt$.

From Figure 32. we could select the operating points given by:

A point: $\alpha = 9, \beta = 0.1$

B point: $\alpha = 20, \beta = 0.2$

C point: $\alpha = 20, \beta = 0.4$

The operating point A ($\alpha = 9.0$, $\beta = 0.1$) provides the minimum cost.

Recall that

$$\alpha = \frac{T1}{T2}$$

$$\beta = \frac{1}{T2}$$

The two parameters values become

$$T1 = 90.0,\ T2 = 10.0$$

Comparison of Reid's result and the simulation result show that they are very closely matched. In addition, we simulated yaw response and rudder activity at the operating points A , B and C in Figure 33. Figure 34 compares the yaw response to course change for the ship at each operating point. Figure 35 compares the rudder activity. It is noted that operating point C is a faster controller but has a higher cost value. This cost is a higher rate of fuel consumption used to overcome the higher hydrodynamic drag forces.

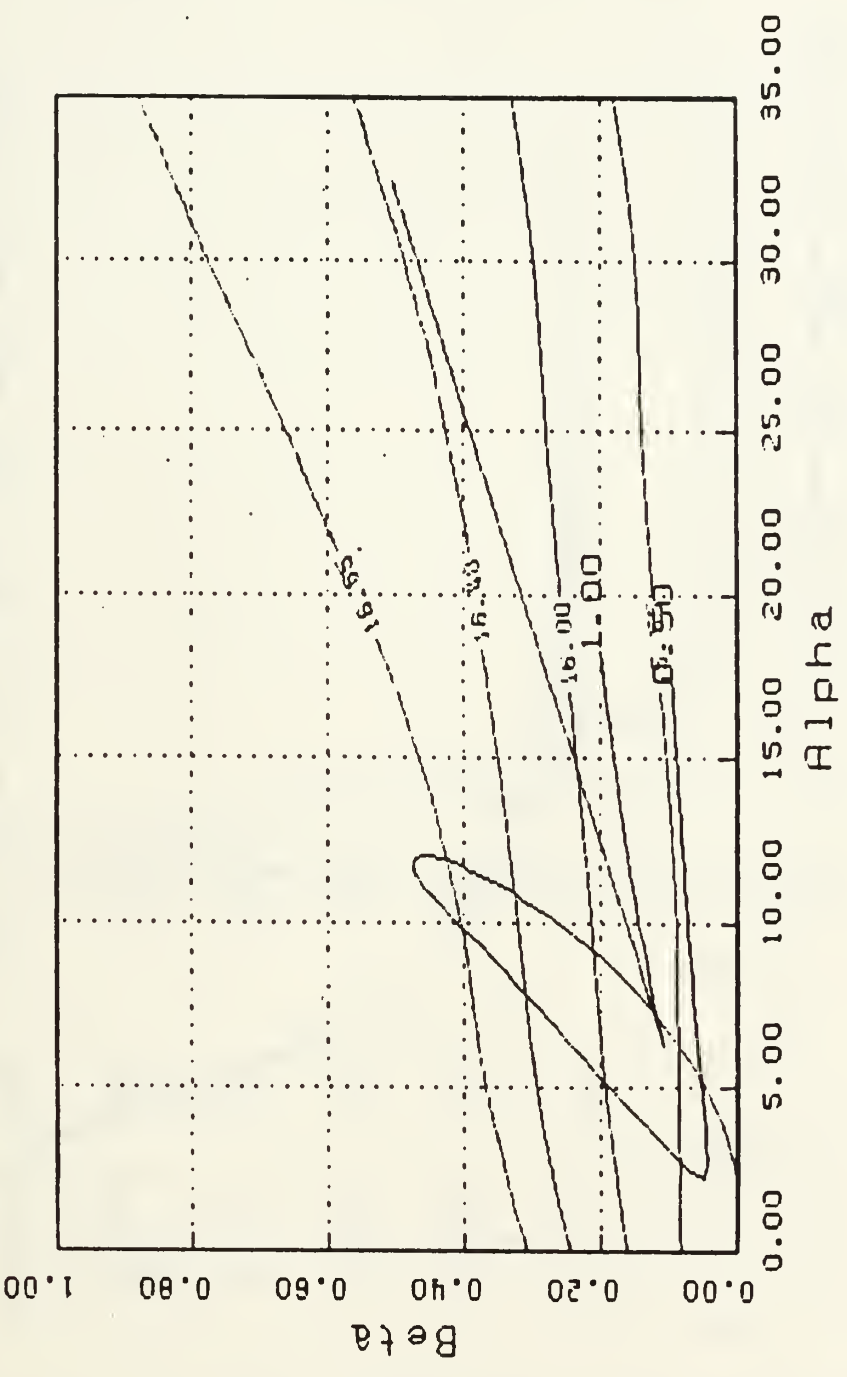


Figure 33. Combine zeta and cost curves for $\frac{1}{T}\int_0^T(\lambda\psi_\epsilon^2 + \delta^2)\,dt$.

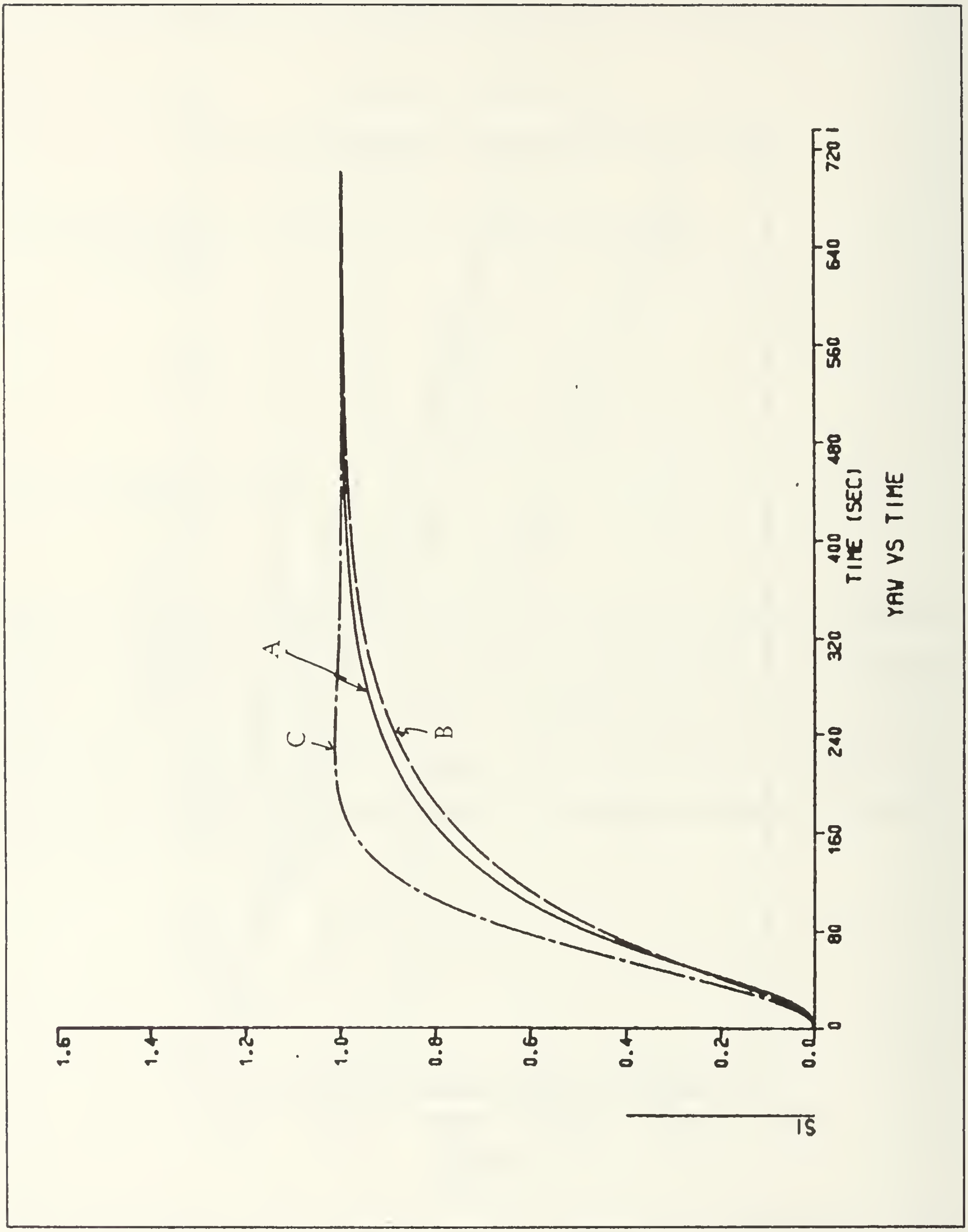


Figure 34. The yaw response vs time.

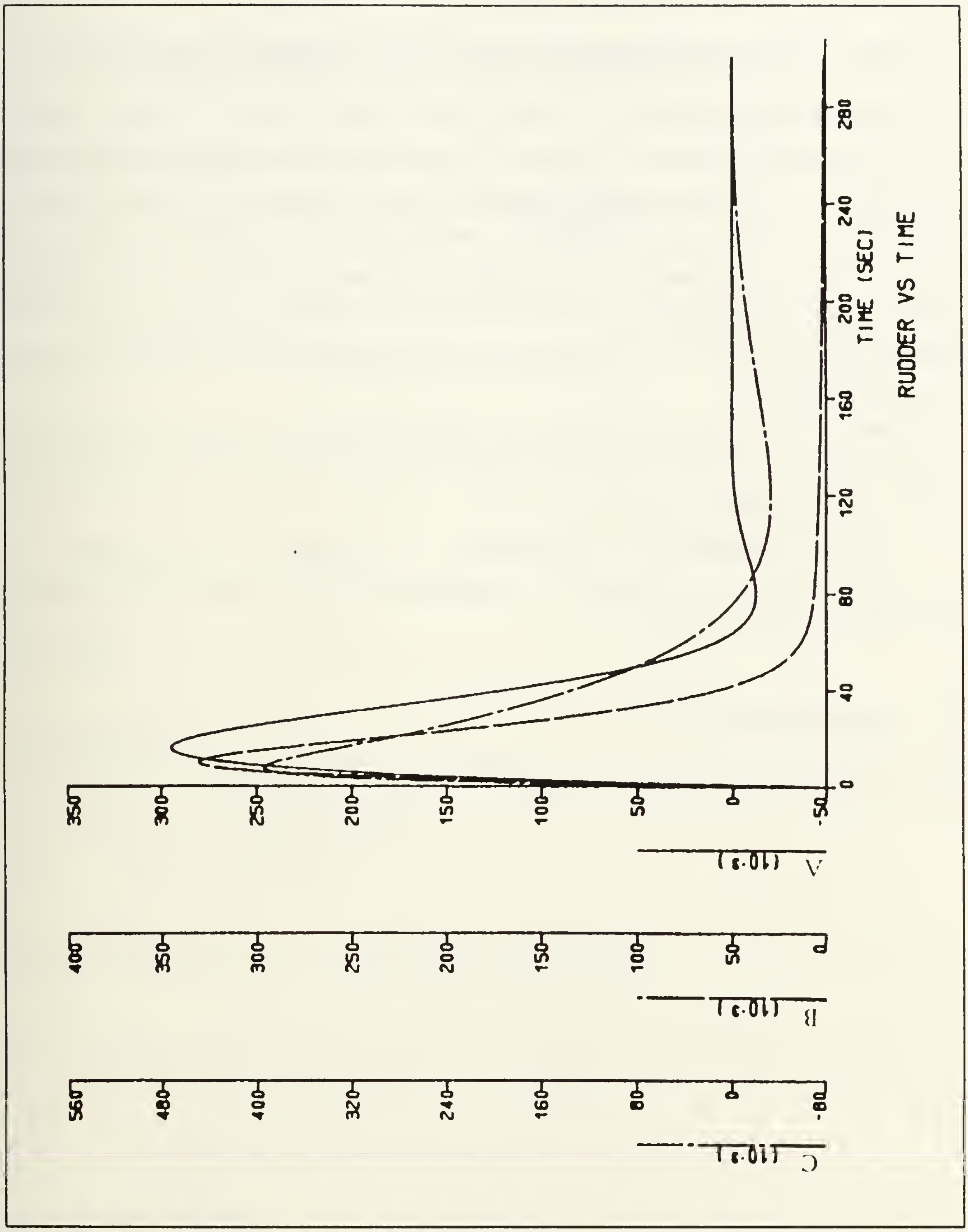


Figure 35. The rudder activity vs time.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The graphical solution of the parameter plane technique provides the designer with a visual means of deducing how the dominant roots of the characteristic equation move about in the s-plane as two user-defined parameters defining compensator attributes are varied. This was illustrated in the example problems in Chapter 4. The graphical method is able to analyze easily, not only the linear constraint $\int E^2 \, dt$ but also the non-linear constraint $\int |E| \, t \, dt$, $\int |E| \, dt$ and any kind of cost function using the performance index program.

Constant cost value contours can be plotted in much the same presentation as the parameter plane plot. This procedure is a useful design tool which permits simultaneous analysis of both dynamic response and cost.

As has been demonstrated, rapid solution of a wide range of linear automatic control systems problems can be achieved through use of the performance index program developed as part of this thesis.

B. RECOMMENDATIONS

Further work should be done to combine individual programs so that both parameter plane and cost curves are produced by one consolidated program.

APPENDIX A. TABULATED VALUES OF THE INTEGRAL FORM

Table A.1 gives the value of I_n for values of n from 1 to 9 where

$$I_n = \frac{1}{2\pi j} \int_{-j\infty}^{+j\infty} ds \, \frac{c(s)c(-s)}{d(s)d(-s)} \qquad (A.1-1)$$

and

$$c(s) = c_{n-1}s^{n-1} + \cdots + c_0 \qquad (A.1-2)$$

$$d(s) = d_n s^n + \cdots + d_0 \qquad (A.1-3)$$

Table A.1

$$I_1 = \frac{c_0^2}{2d_0d_1};$$

$$I_2 = \frac{c_1^2 d_0 + c_0^2 d_2}{2d_0d_1d_2};$$

$$I_3 = \frac{c_2^2 d_0 d_1 + (c_1^2 - 2c_0c_2)d_0d_3 + c_0^2 d_2 d_3}{2d_0d_3(-d_0d_3 + d_1d_2)};$$

$$I_4 = \frac{c_3^2(-d_0^2 d_3 + d_0d_1d_2) + (c_2^2 - 2c_1c_3)d_0d_1d_4 + (c_1^2 - 2c_0c_2)d_0d_3d_4 + c_0^2(-d_1d_4^2 + d_2d_3d_4)}{2d_0d_4(-d_0d_3^2 - d_1^2 d_4 + d_1d_2d_3)};$$

$$I_5 = \frac{1}{2\Delta_5}\left[c_4^2 m_0 + (c_3^2 - 2c_2c_4)m_1 + (c_2^2 - 2c_1c_3 + 2c_0c_4)m_2 + (c_1^2 - 2c_0c_2)m_3 + c_0^2 m_4\right];$$

where

$$m_0 = \frac{1}{d_5}(d_3m_1 - d_1m_2),$$

$$m_1 = -d_0d_3 + d_1d_2,$$

$$m_2 = -d_0 d_5 + d_1 d_4,$$

$$m_3 = \frac{1}{d_0}(d_2 m_2 - d_4 m_1),$$

$$m_4 = \frac{1}{d_0}(d_2 m_3 - d_4 m_2), \text{ and}$$

$$\Delta_5 = d_0(d_1 m_4 - d_3 m_3 + d_5 m_2).$$

$$I_6 = \frac{1}{2\Delta_6}\left[c_5^2 m_0 + (c_4^2 - 2c_3 c_5)m_1 + (c_3^2 - 2c_2 c_4 + 2c_1 c_5)m_2 + (c_2^2 - 2c_1 c_3 + 2c_0 c_4)m_3\right] + \frac{1}{2\Delta_6}\left[(c_1^2 - 2c_0 c_2)m_4 c_0^2 m_5\right];$$

where

$$m_0 = \frac{1}{d_6}(d_4 m_1 - d_2 m_2 + d_0 m_3),$$

$$m_1 = -d_0 d_1 d_5 + d_0 d_3^2 + d_1^2 d_4 - d_1 d_2 d_3,$$

$$m_2 = d_0 d_3 d_5 + d_1^2 d_6 - d_1 d_2 d_5,$$

$$m_3 = d_0 d_5^2 + d_1 d^3 d_6 - d_1 d_4 d_5,$$

$$m_4 = \frac{1}{d_0}(d_2 m_3 - d_4 m_2 + d_6 m_1),$$

$$m_5 = \frac{1}{d_0}(d_2 m_4 - d_4 m_3 + d_6 m_2), \text{ and}$$

$$\Delta_6 = d_0(d_1 m_5 - d_3 m_4 + d_5 m_3).$$

$$I_7 = \frac{1}{2\Delta_7}\left[c_6^2 m_0 + (c_5^2 - 2c_4 c_6)m_1 + (c_4^2 - 2c_3 c_5 + 2c_2 c_6)m_2 + (c_3^2 - 2c_2 c_4 + 2c_1 c_5 - 2c_0 c_6)m_3\right] + \frac{1}{2\Delta_7}\left[(c_2^2 - 2c_1 c_3 + 2c_0 c_4)m_4 + (c_1^2 - 2c_0 c_2)m_5 + c_0^2 m_6\right];$$

where

$$m_0 = \frac{1}{d_7}(d_5m_1 - d_3m_2 + d_1m_3),$$

$$m_1 = -(d_1d_4 - d_0d_5)^2 + (d_0d_3 - d_1d_2)(d_0d_7 - d_1d_6 + d_2d_5 - d_3d_4),$$

$$m_2 = (d_0d_7 - d_1d_6)(-d_0d_5 + d_1d_4) + (d_0d_3 - d_1d_2)(d_2d_7 - d_3d_6),$$

$$m_3 = -(d_0d_7 - d_1d_6)^2 + (d_0d_3 - d_1d_2)(d_5d_7 - d_5d_6),$$

$$m_4 = \frac{1}{d_0}(d_2m_3 - d_4m_2 + d_6m_1),$$

$$m_5 = \frac{1}{d_0}(d_2m_4 - d_4m_3 + d_6m_2),$$

$$m_6 = \frac{1}{d_0}(d_2m_5 - d_4m_4 + d_6m_3), \text{ and}$$

$$\Delta_7 = d_0(d_1m_6 - d_3m_5 + d_5m_4 - d_7m_3).$$

$$I_8 = \frac{1}{2\Delta_8}\left[c_7^2m_0 + (c_6^2 - 2c_5c_7)m_1 + (c_5^2 - 2c_4c_6 + 2c_3c_7)m_2 + (c_4^2 - 2c_3c_5 + 2c_2c_6 - 2c_1c_7)m_3\right]$$
$$+ \frac{1}{2\Delta_8}\left[(c_3 - 2c_2c_4 + 2c_1c_5 - 2c_0c_6)m_4 + (c_2^2 - 2c_1c_3 + 2c_0c_4)m_5 + (c_1^2 - 2c_0c_2)m_6 + c_0^2m_7\right];$$

where

$$m_0 = \frac{1}{d_8}(d_6m_1 - d_4m_2 + d_2m_3 - d_0m_4),$$

$$\begin{aligned} m_1 = {} & (d_0d_7 - d_2d_5)(-d_0d_1d_7 + d_0d_3d_5 + 2d_1^2d_6) + (d_3d_7 - d_5^2)(d_0^2d_5 + d_1d_2^2) \\ & + d_1d_3d_8(d_0d_3 - d_1d_2) - d_1^2d_8(d_0d_5 - d_1d_4) \\ & + (-d_2d_7 + d_3d_6 - d_4d_5)(d_0d_3^2 + d_1^2d_4) - d_1d_6(d_1^2d_6 + 3d_0d_3d_5) \\ & - d_1d_2d_3(d_3d_6 - d_4d_5) + 2d_0d_1d_4d_5^2, \end{aligned}$$

$$\begin{aligned} m_2 = {} & (d_0d_3 - d_1d_2)(d_0d_7^2 - d_1d_5d_8 - d_1d_6d_7 + d_2d_5d_7) \\ & + (d_3d_8 - d_4d_7)(-d_0d_1d_5 + d_0d_3^2 - d_1d_2d_3 + d_1^2d_4) \\ & - d_0d_5d_7(d_0d_5 - d_1d_4) + d_1^2d_8(d_0d_7 - d_1d_6), \end{aligned}$$

$$m_3 = -d_1(d_1d_8 - d_2d_7)^2 + (-d_5d_8 + d_6d_7)(d_0d_1d_5 - d_0d_3^2 + d_1d_2d_3 - d_1^2d_4) + d_0d_7^2(-d_0d_5 + d_1d_4 + d_2d_3) - 2d_0d_1d_7d_8,$$

$$m_4 = (-d_5d_8 + d_6d_7)(2d_0d_1d_7 - d_0d_3d_5 + d_1d_2d_5 - d_1^2d_6) + (-d_3d_8 + d_4d_7)(d_0d_3d_7 - d_1d_2d_7 + d_1^2d_8) - d_0^2d_7^3,$$

$$m_5 = \frac{1}{d_0}(d_2m_4 - d_4m_3 + d_6m_2 - d_8m_1),$$

$$m_6 = \frac{1}{d_0}(d_2m_5 - d_4m_4 + d_6m_3 - d_8m_2),$$

$$m_7 = \frac{1}{d_0}(d_2m_6 - d_4m_5 + d_6m_4 - d_8m_3), \text{ and}$$

$$\Delta_8 = d_0(d_1m_7 - d_3m_6 + d_5m_5 - d_7m_4).$$

$$I_9 = \frac{1}{2\Delta_9}\left[c_8^2m_0 + (c_7^2 - 2c_6c_8)m_1 + (c_6^2 - 2c_5c_7 + 2c_4c_8)m_2 + (c_5^2 - 2c_4c_6 + 2c_3c_7 - 2c_2c_8)m_3\right] + \frac{1}{2\Delta_9}\left[(c_4 - 2c_3c_5 + 2c_2c_6 - 2c_1c_7 + 2c_0c_8)m_4 + (c_3^2 - 2c_2c_4 + 2c_1c_5{}_{-}2c_0c_6)m_5\right] + \frac{1}{2\Delta_9}\left[(c_2^2 - 2c_1c_3 + 2c_0c_4)m_6 + (c_1^2 - 2c_0c_2)m_7 + c_0^2m_8\right];$$

where

$$m_0 = \frac{1}{d_9}(d_7m_1 - d_5m_2 + d_3m_3 - d_1m_4),$$

$$m_1 = a_1(a_1a_{10} - a_2a_9 + a_3a_6 + a_3a_8 + 2a_4a_6 - a_5^2 - a_5a_7 - a_7^2) + a_2(-a_2a_6 - a_3a_7 + a_4a_5 + 2a_4a_7) - a_4^3,$$

$$m_2 = a_1(a_3a_9 + a_4a_9 - a_5a_8 + a_6a_7 - a_7a_8) + a_2(-a_2a_9 + a_4a_8 + a_7^2) - a_4^2a_7,$$

$$m_3 = a_1(a_3a_{10} + a_4a_{10} + a_7a_9 - a_8^2) + a_2(-a_2a_{10} + a_7a_8) - a_4a_7^2,$$

$$m_4 = a_1(a_5a_{10} + 2a_7a_{10} - a_8a_9) + a_2(a_7a_9 - a_4a_{10}) - a_7^3,$$

$$m_5 = \frac{1}{d_0}(d_2m_4 - d_4m_3 + d_6m_2 - d_8m_1).$$

$$m_6 = \frac{1}{d_0}(d_2 m_5 - d_4 m_4 + d_6 m_3 - d_8 m_2),$$

$$m_7 = \frac{1}{d_0}(d_2 m_6 - d_4 m_5 + d_6 m_4 - d_8 m_3),$$

$$m_8 = \frac{1}{d_0}(d_2 m_7 - d_4 m_6 + d_6 m_5 - d_8 m_4), \text{ and}$$

$$\Delta_9 = d_0(d_1 m_8 - d_3 m_7 + d_5 m_6 - d_7 m_5 + d_9 m_4),$$

where

$$a_1 = d_1 d_2 - d_0 d_3$$

$$a_2 = d_1 d_4 - d_0 d_5$$

$$a_3 = d_3 d_4 - d_2 d_5$$

$$a_4 = d_1 d_6 - d_0 d_7$$

$$a_5 = d_3 d_6 - d_2 d_7$$

$$a_6 = d_5 d_6 - d_4 d_7$$

$$a_7 = d_1 d_8 - d_0 d_9$$

$$a_8 = d_3 d_8 - d_2 d_9$$

$$a_9 = d_5 d_8 - d_4 d_9$$

$$a_{10} = d_7 d_8 - d_6 d_9.$$

APPENDIX B. PERFORMANCE INDEX PROGRAM.

This appendix contains a listing file of the FORTRAN source code for the performance index program. This program is used to plot the performance index curves for desired values.

The program consists of main program and subroutines. The subroutine SIZE permits adjustment of plot size to both the screen and the printer. The subroutine LDFILE permits program input of a pre-existing data file. Subroutine MONPRT allows selection of the graphics output device, be it the monitor or a wide variety of printers.

```
* * * * * * ARRAY DEFINITION * * * * * *

      CHARACTER*1 CHAR,CHG
      REAL Z(51,51)
      REAL XP(202),YP(202),ZP(200)
      REAL   ZPIJ(200)
      INTEGER KNXT(200),ITEXT(1)
      INTEGER J,K,SEL
      REAL ZLEV(50)
      INTEGER LDIG(50),LWGT(50)
      COMMON/PLOTTR/ IOPORT,MODEL

 * * * * * * VARIABLE INITIAL VALUE * * * * * *

      DATA XMAX/11.0/,YMAX/8.0/
      DATA XLPLOT/1.0/,YLPLOT/0.5/,XHPLOT/8.0/,YHPLOT/5.5/
      DATA CAY/5.0/,NRNG/3/
      DATA HGT/.12/,NDIV/4/,NARC/8/
      DATA NSM/4/

 * * * * * * ENTER INPUT PARAMETERS * * * * * *

   10 CONTINUE
      CALL CLR
      CALL CUP(1,0)
        MODEL=99
        IOPORT=99
  13  WRITE(*,297)
      WRITE(*,305)
      READ(*,*) NLEV
      IF (NLEV.LT.0.OR.NLEV.GT.50) GO TO 13
      DO 22 K=1,NLEV
      WRITE(*,304) K
      READ(*,*) ZLEV(K),LWGT(K),LDIG(K)
  22  CONTINUE
      WRITE(*,300)
      READ(*,*) NPTS
      WRITE(*,297)

 * * * * * * BEGIN PROCESSING * * * * * *

      FACT=0.9
      CALL SIZE (FACT)

   25 CONTINUE

      CALL LDFILE(NPTS,XP,YP,ZP)

C...  INITIALIZE PLOT88
      CALL PLOTS(0,IOPORT,MODEL)

C...  DEFINE OUTPUT WINDOW
      CALL WINDOW(0.,0.,XMAX,YMAX)

C...  REORIGIN THE ENTIRE DRAWING
      CALL PLOT(1.0,0.5,-3)

      XPMAX=XP(1)
      YPMAX=YP(1)
      XPMIN=XP(1)
      YPMIN=YP(1)
      DO 3 K=2,NPTS
         IF (XP(K).GT.XPMAX) THEN
             XPMAX=XP(K)
         END IF
         IF (YP(K).GT.YPMAX) THEN
             YPMAX=YP(K)
         END IF
         IF (XP(K).LT.XPMIN) THEN
             XPMIN=XP(K)
         END IF
         IF (YP(K).LT.YPMIN) THEN
             YPMIN=YP(K)
         END IF
  3   CONTINUE
      DO 4 J=1,NPTS
         XP(J)=XP(J)*XHPLOT/XPMAX
         YP(J)=YP(J)*YHPLOT/YPMAX
         XHIGH=XHPLOT
         YHIGH=YHPLOT
         XLOW=XPMIN*XHPLOT/XPMAX
         YLOW=YPMIN*YHPLOT/YPMAX
  4   CONTINUE
C...  ZERO GRID ARRAY Z
      DO 200 I=1,51
      DO 200 J=1,51
      Z(I,J)=0.0
 200  CONTINUE

C...  GRID SIZE PARAMETERS
      NX=8
      NY=6
      DX=(XPMAX-XPMIN)/FLOAT(NX-1)
      DY=(YPMAX-YPMIN)/FLOAT(NY-1)

      CALL FACTOR (FACT)
      CALL STAXIS (.13,.20,.15,0.1,2)
      CALL AXIS(1.0,0.5,'Alpha',-5,-7.,0.,XPMIN,DX)
      CALL AXIS(1.0,0.5,'Beta',4,-5.0,90.,YPMIN,DY)
      CALL ZGRID(Z,51,51,NX,NY,XLOW,YLOW,XHIGH,YHIGH,XP,YP,ZP,
     1   NPTS,CAY,NRNG,ZPIJ,KNXT)

C..   POST THE INPUT POINTS.

      CALL ZPOST(XLOW,YLOW,XHIGH,YHIGH,XLPLOT,YLPLOT,XHPLOT,YHPLOT,
     1    XP,YP,ZP,NPTS,2,'(1x,f5.2)   ',.1389)

      IRTN=590

C -------- PUT A BOX AROUND THE MAP --------------

 580  CALL PLOT(XLPLOT,YLPLOT,3)
      CALL PLOT(XHPLOT,YLPLOT,2)
      CALL PLOT(XHPLOT,YHPLOT,2)
      CALL PLOT(XLPLOT,YHPLOT,2)
      CALL PLOT(XLPLOT,YLPLOT,2)

      IF(IRTN.EQ.900) GOTO 900

  590 CALL PLOT(1.0,0.5,-999)
      CALL PLOT(1.0,0.5,-3)
      CALL FACTOR (FACT)
      CALL STAXIS (.13,.20,.15,0.1,2)
      CALL AXIS(1.0,0.5,'Alpha',-5,-7.,0.,XPMIN,DX)
      CALL AXIS(1.0,0.5,'Beta',4,-5.0,90.,YPMIN,DY)

C ------ SMOOTH THE DATA BEFORE DRAWING ------------

      CALL ZSMTH(Z,51,51,NX,NY,NSM)
      CALL FACTOR (FACT)
      CALL STAXIS (.13,.20,.15,0.1,2)
      CALL AXIS(1.0,0.5,'Alpha',-5,-7.,0.,XPMIN,DX)
      CALL AXIS(1.0,0.5,'Beta',4,-5.0,90.,YPMIN,DY)

      CALL ZCSEG(Z,51,51,NX,NY,XLPLOT,YLPLOT,XHPLOT,YHPLOT,
     1 ZLEV,LDIG,LWGT,NLEV,HGT,NDIV,NARC)
      IRTN=900
      GO TO 580
  900 CALL PLOT(1.0,1.0,999)

      WRITE(*,*)'  WOULD YOU LIKE HARDCOPY OF THESE PLOTS? (Y or N)'
      READ(*,401)CHAR
      IF((CHAR.EQ.'Y').OR.(CHAR.EQ.'y')) THEN
        CALL MONPRT
        GOTO 25
      ENDIF

      WRITE(*,*)'  WOULD YOU LIKE TO DO ANOTHER ?  (Y or N)'
      READ(*,401) CHAR
      IF((CHAR.EQ.'Y').OR.(CHAR.EQ.'y')) GOTO 10

  297 FORMAT(1x)
  300 FORMAT('      3.Enter number of data file points ===>', )
  304 FORMAT('      2.Enter PI values ,lwgt,ldig # ',i2,'===>', )
  305 FORMAT('      1.How many constant PI curves # ===>', )
  401 FORMAT(A)

C...  end of program
      STOP
      END

C -------------------------------------------
C     SUBROUTINE CLR                        *
C     SUBROUTINE TO ERASE THE DISPLAY.      *
C     CALL,   CALL CLR                      *
C -------------------------------------------

      SUBROUTINE CLR
      CHARACTER*1 C1,C2,C3,C4
      INTEGER*2 IC(4)
      EQUIVALENCE (C1,IC(1)),(C2,IC(2)),(C3,IC(3)),(C4,IC(4))
      DATA IC/16#1B,16#5B,16#32,16#4A/

C...  WRITE ESCAPE CODE TO DISPLAY
      WRITE(*,1) C1,C2,C3,C4
    1 FORMAT(1x,4a1)
      RETURN
      END
C ---------------------------------------------
C     SUBROUTINE  CUP                         *
C     MOVE THE CURSOR TO LINE N, COLUMN M.    *
C     CALL,  CALL CUP(N,M)                    *
C     N - NEW CURSOR LINE                     *
C     M - NEW CURSOR COLUMN                   *
C ---------------------------------------------

      SUBROUTINE CUP (N,M)
      CHARACTER*1 C1,C2,C5,C8,LC(5)
      CHARACTER*5 CBUFF
      INTEGER*2 IC(4)
      EQUIVALENCE (C1,IC(1)),(C2,IC(2)),(C5,IC(3)),(C8,IC(4)),
     *(CBUFF,LC(1))
      DATA IC/16#1B,16#5B,16#3B,16#66/

      L=10000+100*N+M

      WRITE(CBUFF,2)L
   2  FORMAT(i5)
      WRITE(*,1) C1,C2,LC(2),LC(3),C5,LC(4),LC(5),C8
    1 FORMAT(1x,8a1, )
      RETURN
      END
C -----------------------------------------------------------------------
C                                                                       *
```

```
C  SUBROUTINE  LDFILE -- ALLOWS LOADING OF DATA FROM AN EXISTING FILE   *
C                                                                        *
C -----------------------------------------------------------------------

      SUBROUTINE LDFILE (NPTS,XP,YP,ZP)
      REAL XP(202),YP(202),ZP(200)
      INTEGER NPTS,I
      CHARACTER*12 FILEIN,ANS*1
      LOGICAL EXIST
    1 CALL CLR
      WRITE(*,100)
  100 FORMAT(1X,////,10X,'-------------------------------------------
     *------',/,10X,'*** THIS IS THE ROUTINE TO LOAD DATA FROM A FILE **
     **',/,10X,'-----------------------------------------------------'
     *,//,10X,'FILE NAME SHOULD NOT EXCEED 8 CHARACTERS',/,
     *10X,'FILE EXTENSION SHOULD NOT EXCEED 3 CHARACTERS')
      WRITE(*,101)
  101 FORMAT(1X,////,10X,'WHAT IS THE FILE NAME (FN) AND EXTENSION (EXT)
     *? ',//,10X,'ENTER IN FORM FN.EXT (E.G. PARAPLAN.INP) ===> ')
      CALL CUP(18,57)
      READ(*,102) FILEIN
  102 FORMAT(A12)
      INQUIRE(FILE=FILEIN,EXIST=EXIST)
      IF (EXIST) THEN

        OPEN(7,FILE=FILEIN,STATUS='OLD',ACCESS='SEQUENTIAL')
      ELSE
        WRITE(*,103)
  103   FORMAT(///,10X,'THAT FILE DOES NOT EXIST/,10X,
     *         'DO YOU WANT TO TRY AGAIN?  (Y/N)')
        READ(*,'(A1)') ANS
        IF((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
          GOTO 1
        ELSE
          GOTO 2
        END IF
      END IF
      DO 190 I = 1,NPTS
        READ(7,120) XP(I),YP(I),ZP(I)
120   FORMAT(1X,F10.2,3X,F10.2,3X,F10.3)
190   CONTINUE
      CLOSE(7,STATUS='KEEP')
2     RETURN
      END

C ----------------------------------------------------------------
C                                                                *
C  SUBROUTINE  MONPRT -- ALLOWS DISPLAY ON MONITOR OR PRINTER    *
C                                                                *
C ----------------------------------------------------------------

      SUBROUTINE MONPRT
      INTEGER IOPORT,MODEL,SEL
      COMMON/PLOTTR/ IOPORT,MODEL

      CALL CLR
      WRITE(*,80)
      WRITE(*,81)
      WRITE(*,82)
      WRITE(*,81)
      WRITE(*,83)
      WRITE(*,87)
      WRITE(*,84)
      WRITE(*,81)
      WRITE(*,93)
80    FORMAT(1x,///,10x,'-----------------------------------------------
     *-----------------',/,10x,' ',21x,'PRINTER/OUTPUT MENU',22x,' ')
81    FORMAT(10x,'------------------------------------------------------
     *----------')
87    FORMAT(10x,'++++++++++++++++++++++++++++++++++++++++++++++++++++++
     *+++++++++++')
82    FORMAT(10x,' ',2x,'PRINTER NO.',2x,' ',20x,'PRINTER',19x,' ')
83    FORMAT(10x,' ',3x,'    1     ',2x,' ',4x,'Epson FX-80, All'
     *,26x,' '
     *,/,10x,' ',3x,'    2     ',2x,' ',4x,'Epson FX-100, All'
     *,25x,' '
     *,/,10x,' ',3x,'    3     ',2x,' ',4x,'Epson MX-100, All'
     *,25x,' '
     *,/,10x,' ',3x,'    4     ',2x,' ',4x,'Epson RX-80, All'
     *,26x,' '
     *,/,10x,' ',3x,'    5     ',2x,' ',4x,'Epson MX-80 & IBM Printer'
     *,17x,' '
     *,/,10x,' ',3x,'    6     ',2x,' ',4x,'HP 7470A Graphics Plotter'
     *,17x,' '
     *,/,10x,' ',3x,'    7     ',2x,' ',4x,'HP 7475A Graphics Plotter'
     *,17x,' '
     *,/,10x,' ',3x,'    8     ',2x,' ',4x,'HP 758xB Series Plotters'
     *,18x,' '
     *,/,10x,' ',3x,'    9     ',2x,' ',4x,'HP 2686A Laser Jet'
     *,24x,' ')
84    format(10x,' ',3x,
     *'   10     ',2x,' ',4x,'Graphics Monitor (default)'
     *,16x,' '
     *,/,10x,' ',3x,'   11     ',2x,' ',4x,'HARDWARE Interface Menu'
     *,19x,' '
     *,/,10x,' ',3x,'   12     ',2x,' ',4x,'Input PLOT88 Values for IOPO
     *RT and MODEL'
     *,2x,' '
     *,/,10x,' ',3x,'   99     ',2x,' ',4x,'RETURN to LOAD Data',23x,
     *' ')
93    FORMAT(1x,///,15x,'ENTER INTEGER NUMBER FOR SELECTION ===> ')
      CALL CUP(24,56)
      READ(*,*) SEL
      IF (SEL .EQ. 1) THEN
        IOPORT = 0
        MODEL = 5
      ELSE IF (SEL .EQ. 2) THEN
        IOPORT = 0
        MODEL = 15
      ELSE IF (SEL .EQ. 3) THEN
        IOPORT = 0
        MODEL = 11
      ELSE IF (SEL .EQ. 4) THEN
        IOPORT = 0
        MODEL = 1
      ELSE IF (SEL .EQ. 5) THEN
        IOPORT = 0
        MODEL = 1
      ELSE IF (SEL .EQ. 6) THEN
        IOPORT = 9600
        MODEL = 20
      ELSE IF (SEL .EQ. 7) THEN
        IOPORT = 9600
        MODEL = 30
      ELSE IF (SEL .EQ. 8) THEN
        IOPORT = 9600
        MODEL = 80
      ELSE IF (SEL .EQ. 9) THEN
        IOPORT = 9600
        MODEL = 60
      ELSE IF (SEL .EQ. 11) THEN
        CALL PORT
      ELSE IF (SEL .EQ. 12) THEN
        CALL CLR
        WRITE(*,100)
 100    FORMAT(1x,////,5x,'INPUT VALUE TO BE USED BY THE PLOT88 GRAPHICS
     * PACKAGE',/,5x,'FOR IOPORT ==>',//)
        CALL CUP(7,21)
        READ(*,*) IOPORT
        WRITE(*,110)
 110    FORMAT(1x,///,5x,'INPUT VALUE TO BE USED BY THE PLOT88 GRAPHICS
     *PACKAGE',/,5x,'FOR MODEL ==>',//)
        CALL CUP(12,20)
        READ(*,*) MODEL
      ELSE
        IOPORT = 99
        MODEL = 99
      ENDIF
      WRITE(*,98) IOPORT,MODEL
 98   FORMAT(//,10x,'PLOT88 WILL USE THESE VALUES TO OUTPUT GRAPHICS:'
     *,//,10x,'IOPORT = ',I4,10x,'MODEL = ',I4,///)
      PAUSE
      RETURN
      END

C ---------------------------------------------------------------
C                                                               *
C  SUBROUTINE  PORT -- ALLOWS SELECTION OF OUTPUT PRINTER PORT  *
C                                                               *
C ---------------------------------------------------------------

      SUBROUTINE PORT
      INTEGER IOPORT,MODEL,SEL
      COMMON PLOTTR/ IOPORT,MODEL

      CALL CLR
      WRITE(*,79)
 79   FORMAT(1x,///,15x,'SELECTION OF A PRINTER IN THE PREVIOUS MENU AUT
     *OMATICALLY',/,10x,'SELECTS THE MOST COMMONLY ASSOCIATED OUTPUT POR
     *T, BE IT PARALLEL ',/,10x,
     *'(LPT) OR SERIAL, FOR THAT PARTICULAR DEVICE.  IF THE PROGRAM GRAPH
     *ICS',/,10x,
     *'ARE NOT BEING OUTPUT CORRECTLY, USE THIS MENU TO PROPERLY ROUTE T
     *HE',/,10x,
     *'GRAPHICS DATA TO THE OUTPUT DEVICE.',////)
      PAUSE

      CALL CLR
      WRITE(*,80)
      WRITE(*,81)
      WRITE(*,82)
      WRITE(*,81)
      WRITE(*,83)
      WRITE(*,87)
      WRITE(*,84)
      WRITE(*,81)
      WRITE(*,93)
80    FORMAT(1x,///,10x,'-----------------------------------------------
     *-----------------',/,10x,' ',19x,'HARDWARE INTERFACE MENU'
     *,20x,' ')
81    FORMAT(10x,'------------------------------------------------------
     *----------')
87    FORMAT(10x,'++++++++++++++++++++++++++++++++++++++++++++++++++++++
     *++++++++++')
82    FORMAT(10x,' ',2x,'SELECT NO.',3x,' ',20x,'PORT',22x,' ')
83    FORMAT(10x,' ',3x,'    1     ',2x,' ',14x,'LPT1 Printer Port'
     *,15x,' '
     *,/,10x,' ',3x,'    2     ',2x,' ',14x,'LPT2 Printer Port'
     *,15x,' '
     *,/,10x,' ',3x,'    3     ',2x,' ',14x,'LPT3 Printer Port'
     *,15x,' '
     *,/,10x,' ',3x,'    4     ',2x,' ',14x,'COM1 Serial Port'
     *,16x,' '
     *,/,10x,' ',3x,'    5     ',2x,' ',14x,'COM2 Serial Port'
     *,16x,' ')
84    FORMAT(10x,' ',3x,
     *'   99     ',2x,' ',14x,'RETURN to LOAD Data',13x,' ')
93    FORMAT(1x,///,15x,'ENTER INTEGER NUMBER FOR SELECTION ===> ')
      CALL CUP(21,56)
      READ(*,*) SEL
      IF (SEL .EQ. 1) THEN
        IOPORT = 0
      ELSE IF (SEL .EQ. 2) THEN
        IOPORT = 2
      ELSE IF (SEL .EQ. 3) THEN
        IOPORT = 3
      ELSE IF (SEL .EQ. 4) THEN
        IOPORT = 0
      ELSE IF (SEL .EQ. 5) THEN
        IOPORT = 50
      ELSE
        IOPORT = 99
      ENDIF
      IF ((SEL .EQ. 4) .OR. (SEL .EQ. 5)) THEN
```

```
       CALL CLR
       WRITE(*,60)
       WRITE(*,61)
       WRITE(*,62)
       WRITE(*,61)
       WRITE(*,63)
       WRITE(*,61)
       WRITE(*,93)
60     FORMAT(1x,///,10x,'------------------------------------------------
    *-----------------',/,10x,' ',16x,'BAUD (data transfer) RATE MENU'
    *,16x,' ')
61     FORMAT(10x,'-------------------------------------------------------
    *----------')
62     FORMAT(10x,' ',2x,'SELECT NO.',3x,' ',15x,'BAUD RATE',22x,' ')
63     FORMAT(10x,' ',3x,'    1     ',2x,' ',18x,' 300'
    *,24x,' '
    *,/,10x,' ',3x,'    2     ',2x,' ',18x,'1200'
    *,24x,' '
    *,/,10x,' ',3x,'    3     ',2x,' ',18x,'4800'
    *,24x,' '
    *,/,10x,' ',3x,'    4     ',2x,' ',18x,'9600'
    *,24x,' ')
     CALL CUP(18,56)
     READ(*,*) SEL
     IF (SEL .EQ. 1) THEN
       IOPORT = IOPORT + 300
     ELSE IF (SEL .EQ. 2) THEN
       IOPORT = IOPORT + 1200
     ELSE IF (SEL .EQ. 3) THEN
       IOPORT = IOPORT + 4800
     ELSE
       IOPORT = IOPORT + 9600
     ENDIF

       CALL CLR
       WRITE(*,50)
       WRITE(*,51)
       WRITE(*,52)
       WRITE(*,51)
       WRITE(*,53)
       WRITE(*,51)
       WRITE(*,93)
50     FORMAT(1x,///,10x,'------------------------------------------------
    *-----------------',/,10x,' ',25x,'PARITY MENU'
    *,26x,' ')
51     FORMAT(10x,'-------------------------------------------------------
    *----------')
52     FORMAT(10x,' ',2x,'SELECT NO.',3x,' ',17x,'PARITY',23x,' ')
53     FORMAT(10x,' ',3x,'    1     ',2x,' ',14x,' NO Parity'
    *,22x,' '
    *,/,10x,' ',3x,'    2     ',2x,' ',14x,'EVEN Parity'
    *,21x,' '
    *,/,10x,' ',3x,'    3     ',2x,' ',14x,'ODD Parity'
    *,22x,' ')
     CALL CUP(17,56)
```

```
      READ(*,*) SEL
      END IF
      IF (SEL .EQ. 3) THEN
        IOPORT = IOPORT + 1
      ELSE IF (SEL .EQ. 2) THEN
        IOPORT = IOPORT + 2
      EENDIF
 99   RETURN
      END

C -----------------------------------------------------
C                                                     *
C  SUBROUTINE  SIZE -- ALLOWS ADJUST OF OUTPUT PLOT   *
C                                                     *
C -----------------------------------------------------

      SUBROUTINE SIZE (FACT)
      REAL FACT
C. . .  ADJUST SIZE OF OUTPUT PLOT
      CALL CLR
      WRITE(*,15)
 15   FORMAT(1x,////,10x,'----------------------------------------------
     *------',/,10x,'*** This is the routine to Adjust the plot size  **
     **',/,10x,'----------------------------------------------------'
     *///,10x,'Output plots are currently sized to fill the console scre
     *en.',//,
     *       10x,'Would you like to adjust the plot size?',/,10x,
     *       'Enter "y" or "n" ==> ')
      CALL CUP(14,31)
      READ(*,'(A)') CHG
      IF ((CHG .EQ. 'n') .OR. (CHG .EQ. 'N')) THEN
        GO TO 17
      END IF
      WRITE(*,16)
 16   FORMAT(///,10x,'A default plot factor size of 0.9 is used to fill
     *the console screen.',/,10x,
     *'This plot size will be halved by entering a value of 0.45.',/,
     *10x,'It will be doubled with a factor entry of 1.8.',//,10x,
     *'Enter factor size ==>  ')
      CALL CUP(22,32)
      READ(*,*) FACT
 17   RETURN
      END
```

APPENDIX C. D.S.L SIMULATION PROGRAM AND DATA FILE.

This appendix contains three listing files and its resulting data files for the D.S.L simulation programs used in examples 4-1, 4-2 and 4-3. These programs are used to produce the data files which generate the plots of the performance index curves on the parameter plane.

A. SIMULATION PROGRAM FOR EXAMPLE 4-1.

```
TITLE PERFORMANCE INDEX CALCULATION
ARRAY Z(1),P(4),C(2),D(2)
TABLE Z(1)=100.,P(1-4)=1.0,11.0,10.0,0.0,D(1)=1.0
PARAM A=0.0,B=0.0,Y0=0.0
INITIAL
      C(1)=A
      C(2)=B
      D(2)=B
      BETA=B
      ALPHA=A
DYNAMIC
      R=STEP(0.)
DERIVATIVE
      Y1=TRNFR(0,3,Y0,Z,P,E)
      M=TRNFR(1,1,Y0,C,D,Y1)
      E=R-M
      ABE=ABS(E)
      PI1=INTGRL(Y0,E*E)
      PI2=INTGRL(Y0,ABE*TIME)
      PI3=INTGRL(Y0,ABE)
CONTROL FINTIM=15,DELT=0.01
PRINT 15.,ALPHA,BETA,PI1,PI2,PI3
END
PARAM A=4.,B=10.
END
PARAM A=5.,B=7.5
END
PARAM A=5.,B=10.
END
PARAM A=5.,B=15
END
PARAM A=6.,B=10
END
PARAM A=7.,B=30
END
PARAM A=7.,B=40
END
PARAM A=7.5,B=15
END
PARAM A=7.5,B=20.
```

```
END
PARAM A=7.5,B=25.
END
PARAM A=7.5,B=35
END
PARAM A=7.5,B=45.
END
PARAM A=7.5,B=50.
END
PARAM A=10.,B=10.
END
PARAM A=10.,B=15
END
PARAM A=10.,B=20.
END
PARAM A=10.,B=25.
END
PARAM A=10.,B=35
END
PARAM A=10.,B=40.
END
PARAM A=12.5,B=10.
END
PARAM A=12.5,B=15
END
PARAM A=12.5,B=20.
END
PARAM A=12.5,B=25.
END
PARAM A=12.5,B=35
END
PARAM A=12.5,B=45.
END
PARAM A=12.5,B=50.
END
PARAM A=15.,B=10.
END
PARAM A=15.,B=15
END
PARAM A=15.,B=20.
END
PARAM A=15.,B=25.
END
PARAM A=15.,B=30.
END
PARAM A=15.,B=35
END
PARAM A=15.,B=40.
END
PARAM A=15.,B=45
END
PARAM A=15.,B=50.
END
PARAM A=17.5,B=15
END
PARAM A=17.5,B=20.
```

```
END
PARAM A=17.5,B=25.
END
PARAM A=17.5,B=35
END
PARAM A=17.5,B=40.
END
PARAM A=20.,B=15
END
PARAM A=20.,B=20.
END
PARAM A=20.,B=25.
END
PARAM A=20.,B=35
END
PARAM A=20.,B=40.
END
PARAM A=20.,B=30.
END
PARAM A=20.,B=60.
END
PARAM A=20.,B=90
END
PARAM A=25.,B=15
END
PARAM A=25.,B=20.
END
PARAM A=25.,B=25.
END
PARAM A=25.,B=35
END
PARAM A=25.,B=40.
END
PARAM A=25.,B=30.
END
PARAM A=30.,B=20.
END
PARAM A=30.,B=25.
END
PARAM A=30.,B=30
END
PARAM A=30.,B=35.
END
PARAM A=30.,B=40.
END
PARAM A=50.,B=20.
END
PARAM A=50.,B=25.
END
PARAM A=50.,B=30
END
PARAM A=50.,B=35.
END
PARAM A=50.,B=40.
END
PARAM A=50.,B=50.
```

```
END
PARAM A=50.,B=60.
END
PARAM A=60.,B=30.
END
PARAM A=60.,B=70
END
PARAM A=60.,B=90.
END
PARAM A=80.,B=30.
END
PARAM A=80.,B=50.
END
PARAM A=80.,B=70.
END
PARAM A=80.,B=90.
END
PARAM A=100.,B=30.
END
PARAM A=100.,B=40.
END
PARAM A=100.,B=50.
END
PARAM A=100.,B=60.
END
PARAM A=100.,B=80.
END
PARAM A=100.,B=100.
END
PARAM A=120.,B=35.
END
PARAM A=120.,B=50.
END
PARAM A=130.,B=35.
END
PARAM A=130.,B=50.
END
PARAM A=140.,B=35.
END
PARAM A=140.,B=50.
END
STOP
```

B. DATA FILE FOR EXAMPLE 4-1.

ALPHA	BETA	IES	IAET	IAE
0.0000	0.000	1.00000	1.50000	1.00000
4.0000	10.000	0.32856	0.47462	0.64140
5.0000	7.500	0.30910	0.53308	0.64346
5.0000	10.000	0.28481	0.33893	0.54590
5.0000	15.000	0.30855	0.42223	0.60431
6.0000	10.000	0.26177	0.30259	0.50802
7.0000	30.000	0.33793	0.66355	0.71529
7.0000	40.000	0.41387	1.1865	0.92510
7.5000	15.000	0.23340	0.18931	0.42445
7.5000	20.000	0.25469	0.25123	0.47959
7.5000	25.000	0.28477	0.38517	0.56794
7.5000	35.000	0.35289	0.76717	0.76028
7.5000	45.000	0.42382	1.27460	0.95456
7.5000	50.000	0.45935	1.57120	1.05080
10.000	10.000	0.25001	0.42101	0.54982
10.000	15.000	0.20401	0.15825	0.37735
10.000	20.000	0.21075	0.14365	0.37559
10.000	25.000	0.22905	0.19092	0.42358
10.000	35.000	0.27603	0.38136	0.55835
10.000	40.000	0.30143	0.50587	0.62925
12.500	10.000	0.28031	0.66678	0.66477
12.500	15.000	0.19186	0.16546	0.37150
12.500	20.000	0.18744	0.11826	0.33404
12.500	25.000	0.19806	0.12439	0.35079
12.500	45.000	0.27154	0.38180	0.55436
12.500	50.000	0.29198	0.47837	0.61059
15.000	10.000	0.34726	1.23910	0.87889
15.000	15.000	0.18889	0.18998	0.38491
15.000	20.000	0.17409	0.11261	0.31716
15.000	25.000	0.17885	0.10424	0.31487
15.000	30.000	0.18996	0.11504	0.33643

15.000	35.000	0.20385	0.14462	0.37180
15.000	40.000	0.21918	0.19111	0.41479
15.000	45.000	0.23531	0.24730	0.45996
15.000	50.000	0.25193	0.31143	0.50594
17.500	15.000	0.19226	0.23148	0.41186
17.500	20.000	0.16644	0.11377	0.31237
17.500	25.000	0.16621	0.09545	0.29500
17.500	35.000	0.18434	0.10984	0.32730
17.500	40.000	0.19669	0.13387	0.35790
20.000	15.000	0.20112	0.29311	0.45121
20.000	20.000	0.16250	0.12059	0.31537
20.000	25.000	0.15766	0.09084	0.28397
20.000	35.000	0.17018	0.09490	0.29932
20.000	40.000	0.18022	0.10688	0.32112
20.000	30.000	0.16203	0.08883	0.28474
20.000	60.000	0.22763	0.23630	0.44709
20.000	90.000	0.30424	0.56953	0.65381
25.000	15.000	0.23759	0.51682	0.57663
25.000	20.000	0.16216	0.15589	0.33829
25.000	25.000	0.14800	0.09050	0.27819
25.000	35.000	0.15137	0.07941	0.26559
25.000	40.000	0.15796	0.08447	0.27703
25.000	30.000	0.14718	0.07807	0.26276
30.000	20.000	0.16949	0.21232	0.37707
30.000	25.000	0.14448	0.10924	0.28776
30.000	30.000	0.13889	0.07458	0.25551
30.000	35.000	0.13990	0.06935	0.24805
30.000	40.000	0.14392	0.07283	0.25209
50.000	20.000	0.31020	1.10170	0.81409
50.000	25.000	0.16519	0.26752	0.39313
50.000	30.000	0.13466	0.15155	0.29655
50.000	35.000	0.12415	0.10402	0.25509
50.000	40.000	0.12067	0.07715	0.23311
50.000	50.000	0.12200	0.05161	0.21576
50.000	60.000	0.12792	0.05862	0.22270

60.000	30.000	0.14256	0.20934	0.33390
60.000	70.000	0.12363	0.05488	0.21479
60.000	90.000	0.13784	0.07347	0.24487
80.000	30.000	0.17846	0.38818	0.45147
80.000	50.000	0.10841	0.10385	0.22883
80.000	70.000	0.10938	0.05337	0.19932
80.000	90.000	0.11809	0.05032	0.20455
100.00	30.000	0.27014	0.88639	0.71319
100.00	40.000	0.12869	0.23817	0.31758
100.00	50.000	0.10764	0.14872	0.24754
100.00	60.000	0.10224	0.10517	0.21883
100.00	80.000	0.10374	0.06032	0.19541
100.00	100.000	0.11050	0.04206	0.19150
120.00	35.000	0.20936	0.58018	0.54931
120.00	50.000	0.11045	0.19635	0.26939
130.00	35.000	0.25186	0.80858	0.66742
130.00	50.000	0.11297	0.22142	0.28185
140.00	35.000	0.32030	1.26700	0.85656
140.00	50.000	0.11619	0.24697	0.29524

C. SIMULATION PROGRAM FOR EXAMPLE 4-2.

```
TITLE PERFORMANCE INDEX CALCULATION
ARRAY A(1),B(4)
TABLE A(1)=50.,B(1)=1.0,B(4)=0.0
PARAM K1=0.2,K2=0.0,Y0=0.0
INITIAL
      B(2)=3+50*K2
      B(3)=2+50*K1
      BETA=K2
      ALPHA=K1
DYNAMIC
      R=STEP(0.)
DERIVATIVE
      M=TRNFR(0,3,Y0,A,B,E)
      E=R-M
      ABE=ABS(E)
      PI1=INTGRL(Y0,E*E)
      PI2=INTGRL(Y0,ABE*TIME)
      PI3=INTGRL(Y0,ABE)
CONTROL FINTIM=15,DELT=0.01
PRINT 15.,ALPHA,BETA,PI1,PI2,PI3
END
PARAM K1=0.2,K2=0.1
END
PARAM K1=0.2,K2=0.2
END
PARAM K1=0.2,K2=0.3
END
PARAM K1=0.2,K2=0.4
END
PARAM K1=0.2,K2=0.5
END
PARAM K1=0.2,K2=0.6
END
PARAM K1=0.2,K2=0.7
END
PARAM K1=0.2,K2=0.8
END
PARAM K1=0.2,K2=0.9
END
PARAM K1=0.2,K2=1.0
END
PARAM K1=0.3,K2=0.1
END
PARAM K1=0.3,K2=0.2
END
PARAM K1=0.3,K2=0.3
END
PARAM K1=0.3,K2=0.4
END
PARAM K1=0.3,K2=0.5
END
PARAM K1=0.3,K2=0.6
END
PARAM K1=0.3,K2=0.7
```

```
END
PARAM K1=0.3,K2=0.8
END
PARAM K1=0.3,K2=0.9
END
PARAM K1=0.3,K2=1.0
END
PARAM K1=0.4,K2=0.1
END
PARAM K1=0.4,K2=0.2
END
PARAM K1=0.4,K2=0.3
END
PARAM K1=0.4,K2=0.4
END
PARAM K1=0.4,K2=0.5
END
PARAM K1=0.4,K2=0.6
END
PARAM K1=0.4,K2=0.7
END
PARAM K1=0.4,K2=0.8
END
PARAM K1=0.4,K2=0.9
END
PARAM K1=0.5,K2=0.1
END
PARAM K1=0.5,K2=0.2
END
PARAM K1=0.5,K2=0.3
END
PARAM K1=0.5,K2=0.4
END
PARAM K1=0.5,K2=0.5
END
PARAM K1=0.5,K2=0.6
END
PARAM K1=0.5,K2=0.7
END
PARAM K1=0.5,K2=0.8
END
PARAM K1=0.5,K2=0.9
END
PARAM K1=0.5,K2=1.0
END
PARAM K1=0.6,K2=0.1
END
PARAM K1=0.6,K2=0.2
END
PARAM K1=0.6,K2=0.3
END
PARAM K1=0.6,K2=0.4
END
PARAM K1=0.6,K2=0.5
END
PARAM K1=0.6,K2=0.6
```

```
END
PARAM K1=0.6,K2=0.7
END
PARAM K1=0.6,K2=0.8
END
PARAM K1=0.6,K2=0.9
END
PARAM K1=0.6,K2=1.0
END
PARAM K1=0.7,K2=0.1
END
PARAM K1=0.7,K2=0.2
END
PARAM K1=0.7,K2=0.3
END
PARAM K1=0.7,K2=0.4
END
PARAM K1=0.7,K2=0.5
END
PARAM K1=0.7,K2=0.6
END
PARAM K1=0.7,K2=0.7
END
PARAM K1=0.7,K2=0.8
END
PARAM K1=0.7,K2=0.9
END
PARAM K1=0.7,K2=1.0
END
PARAM K1=0.8,K2=0.1
END
PARAM K1=0.8,K2=0.2
END
PARAM K1=0.8,K2=0.3
END
PARAM K1=0.8,K2=0.4
END
PARAM K1=0.8,K2=0.5
END
PARAM K1=0.8,K2=0.6
END
PARAM K1=0.8,K2=0.7
END
PARAM K1=0.8,K2=0.8
END
PARAM K1=0.8,K2=0.9
END
PARAM K1=0.8,K2=1.0
END
PARAM K1=0.9,K2=0.1
END
PARAM K1=0.9,K2=0.2
END
PARAM K1=0.9,K2=0.3
END
PARAM K1=0.9,K2=0.4
```

```
END
PARAM K1=0.9,K2=0.5
END
PARAM K1=0.9,K2=0.6
END
PARAM K1=0.9,K2=0.7
END
PARAM K1=0.9,K2=0.8
END
PARAM K1=0.9,K2=0.9
END
PARAM K1=0.9,K2=1.0
END
PARAM K1=1.0,K2=0.1
END
PARAM K1=1.0,K2=0.2
END
PARAM K1=1.0,K2=0.3
END
PARAM K1=1.0,K2=0.3
END
PARAM K1=1.0,K2=0.4
END
PARAM K1=1.0,K2=0.5
END
PARAM K1=1.0,K2=0.6
END
PARAM K1=1.0,K2=0.7
END
PARAM K1=1.0,K2=0.8
END
PARAM K1=1.0,K2=0.9
END
PARAM K1=1.6,K2=1.0
END

STOP
```

D. DATA FILE FOR EXAMPLE 4-2.

ALPHA	BETA	IES	IAET	IAE
0.20000	0.00000	1823.40	1177.4	101.49
0.20000	0.10000	0.81565	4.9245	1.8653
0.20000	0.20000	0.91714	6.0067	2.0813
0.20000	0.30000	1.09550	8.5450	2.5026
0.20000	0.40000	1.28850	11.4640	2.9349
0.20000	0.50000	1.48580	14.6030	3.3547
0.20000	0.60000	1.67890	17.1180	3.3763
0.20000	0.70000	1.87450	20.2920	4.7094
0.20000	0.80000	2.06100	22.7820	4.3853
0.20000	0.90000	2.23130	24.4710	4.6228
0.20000	1.00000	2.40750	26.7370	4.8862
0.30000	0.10000	0.54210	1.3228	1.0492
0.30000	0.20000	0.66416	2.3429	1.3567
0.30000	0.30000	0.80281	3.8884	1.7132
0.30000	0.40000	0.94561	5.7563	2.0709
0.30000	0.50000	1.09000	7.8763	2.4222
0.30000	0.60000	1.23450	9.9336	2.7476
0.30000	0.70000	1.37940	12.3130	3.0747
0.30000	0.80000	1.52230	14.3720	3.3626
0.30000	0.90000	1.66200	16.1900	3.6201
0.40000	0.10000	0.47397	0.5706	0.7763
0.40000	0.20000	0.57805	1.2171	1.0504
0.40000	0.30000	0.68821	2.1595	1.3389
0.40000	0.40000	0.80004	3.3445	1.6271
0.40000	0.50000	0.91257	4.7404	1.9125
0.40000	0.60000	1.02540	6.2509	2.1895
0.40000	0.70000	1.13840	7.9484	2.4640
0.40000	0.80000	1.25100	9.5631	2.7191
0.40000	0.90000	1.36320	11.1670	2.9616
0.40000	1.00000	1.47550	13.0080	3.2087
0.50000	0.10000	0.46277	0.3224	0.6638

0.50000	0.20000	0.55073	0.7599	0.9005
0.50000	0.30000	0.64156	1.3845	1.1390
0.50000	0.40000	0.73322	2.1813	1.3762
0.50000	0.50000	0.82524	3.1372	1.6120
0.50000	0.60000	0.91744	4.2318	1.8455
0.50000	0.70000	1.00970	5.4501	2.0760
0.50000	0.80000	1.10200	6.7009	2.2984
0.50000	0.90000	1.19420	8.0218	2.5160
0.50000	1.00000	1.28650	9.4955	2.7345
0.60000	0.10000	0.47534	0.2533	0.6421
0.60000	0.20000	0.55087	0.5490	0.8301
0.60000	0.30000	0.62798	0.9905	1.0287
0.60000	0.40000	0.70557	1.5579	1.2275
0.60000	0.50000	0.78336	2.2462	1.4262
0.60000	0.60000	0.86125	3.0514	1.6245
0.60000	0.70000	0.93921	3.9528	1.8211
0.60000	0.80000	1.01720	4.9258	2.0147
0.60000	0.90000	1.09520	5.9872	2.2067
0.60000	1.00000	1.17320	7.1377	2.3975
0.70000	0.10000	0.50008	0.3878	0.7400
0.70000	0.20000	0.56605	0.4620	0.8138
0.70000	0.30000	0.63298	0.7805	0.9736
0.70000	0.40000	0.70021	1.2014	1.1408
0.70000	0.50000	0.76757	1.7177	1.3100
0.70000	0.60000	0.83499	2.3256	1.4801
0.70000	0.70000	0.90245	3.0156	1.6500
0.70000	0.80000	0.96993	3.7809	1.8192
0.70000	0.90000	1.03740	4.6278	1.9882
0.70000	1.00000	1.10490	5.5291	2.1552
0.80000	0.10000	0.53189	0.5459	0.8400
0.80000	0.20000	0.59036	0.4757	0.8496
0.80000	0.30000	0.64946	0.6791	0.9597
0.80000	0.40000	0.70875	0.9953	1.0967
0.80000	0.50000	0.76813	1.3928	1.2406
0.80000	0.60000	0.82756	1.8649	1.3873

0.80000	0.70000	0.88701	2.4071	1.5352
0.80000	0.80000	0.94648	3.0174	1.6838
0.80000	0.90000	1.00600	3.6918	1.8324
0.80000	1.00000	1.06540	4.4141	1.9801
0.90000	0.10000	0.56816	0.7239	0.9400
0.90000	0.20000	0.62062	0.6242	0.9400
0.90000	0.30000	0.67352	0.6623	0.9825
0.90000	0.40000	0.72655	0.8877	1.0862
0.90000	0.50000	0.77964	1.1953	1.2059
0.90000	0.60000	0.83276	1.5688	1.3317
0.90000	0.70000	0.88590	2.0029	1.4606
0.90000	0.80000	0.93905	2.4950	1.5913
0.90000	0.90000	0.99221	3.0396	1.7226
0.90000	1.00000	1.04540	3.6298	1.8540
1.0000	0.10000	0.60743	0.9219	1.0400
1.0000	0.20000	0.65498	0.8218	1.0400
1.0000	0.30000	0.70284	0.7427	1.0448
1.0000	0.30000	0.70284	0.7427	1.0448
1.0000	0.40000	0.75080	0.8598	1.1062
1.0000	0.50000	0.79880	1.0886	1.1999
1.0000	0.60000	0.84683	1.3838	1.3055
1.0000	0.70000	0.89487	1.7344	1.4169
1.0000	0.80000	0.94292	2.1352	1.5314
1.0000	0.90000	0.99097	2.5824	1.6476
1.6000	1.00000	1.14690	2.0727	1.7147

E. SIMULATION PROGRAM FOR EXAMPLE 4-3.

```
TITLE PERFORMANCE INDEX CALCULATION
ARRAY A(1),B(3),C(2),D(2)
TABLE A(1)=0.1084,B(1-3)=90.36,1.0,0.0,D(1)=1.0
PARAM Z=0.0,P=0.0,Y0=0.0
CONST K1=0.454616,R=16.296
INITIAL
      C(1)=Z
      C(2)=P
      D(2)=P
      BETA=P
      ALPHA=Z
DYNAMIC
      R=STEP(0.)
DERIVATIVE
      SI=TRNFR(0,2,Y0,A,B,DEL)
      DEL=TRNFR(1,1,Y0,C,D,X1)
      SIE=SI-R
      X1=K1*SIE
      SIES=SIE*SIE
      DELS=DEL*DEL
      Q=R*SIES+DELS
      J=INTGRL(Y0,Q)
CONTROL FINTIM=15.,DELT=0.05
PRINT 15.,ALPHA,BETA,J
END
PARAM Z=2.,P=0.1
END
PARAM Z=2.,P=0.3
END
PARAM Z=2.,P=0.5
END
PARAM Z=2.,P=0.7
END
PARAM Z=2.,P=0.9
END
PARAM Z=2.,P=1.0
END
PARAM Z=4.5,P=0.1
END
PARAM Z=4.5,P=0.3
END
PARAM Z=4.5,P=0.5
END
PARAM Z=4.5,P=0.7
END
PARAM Z=4.5,P=0.9
END
PARAM Z=4.5,P=1.0
END
PARAM Z=7.5,P=0.1
END
PARAM Z=7.5,P=0.3
END
PARAM Z=7.5,P=0.5
```

```
END
PARAM Z=7.5,P=0.7
END
PARAM Z=7.5,P=0.9
END
PARAM Z=7.5,P=1.0
END
PARAM Z=9.,P=0.1
END
PARAM Z=9.,P=0.3
END
PARAM Z=9.,P=0.5
END
PARAM Z=9.,P=0.7
END
PARAM Z=9.,P=0.9
END
PARAM Z=9.,P=1.0
END
PARAM Z=13.,P=0.1
END
PARAM Z=13.,P=0.3
END
PARAM Z=13.,P=0.5
END
PARAM Z=13.,P=0.7
END
PARAM Z=13.,P=0.9
END
PARAM Z=13.,P=1.0
END
PARAM Z=16.5,P=0.1
END
PARAM Z=16.5,P=0.3
END
PARAM Z=16.5,P=0.5
END
PARAM Z=16.5,P=0.7
END
PARAM Z=16.5,P=0.9
END
PARAM Z=16.5,P=1.0
END
PARAM Z=19.,P=0.1
END
PARAM Z=19.,P=0.3
END
PARAM Z=19.,P=0.5
END
PARAM Z=19.,P=0.7
END
PARAM Z=19.,P=0.9
END
PARAM Z=19.,P=1.0
END
PARAM Z=23.5,P=0.1
```

```
END
PARAM Z=23.5,P=0.3
END
PARAM Z=23.5,P=0.5
END
PARAM Z=23.5,P=0.7
END
PARAM Z=23.5,P=0.9
END
PARAM Z=23.5,P=1.0
END
PARAM Z=26.,P=0.1
END
PARAM Z=26.,P=0.3
END
PARAM Z=26.,P=0.5
END
PARAM Z=26.,P=0.7
END
PARAM Z=26.,P=0.9
END
PARAM Z=26.,P=1.0
END
PARAM Z=29.5,P=0.1
END
PARAM Z=29.5,P=0.3
END
PARAM Z=29.5,P=0.5
END
PARAM Z=29.5,P=0.7
END
PARAM Z=29.5,P=0.9
END
PARAM Z=29.5,P=1.0
END
PARAM Z=35.,P=0.1
END
PARAM Z=35.,P=0.3
END
PARAM Z=35.,P=0.5
END
PARAM Z=35.,P=0.7
END
PARAM Z=35.,P=0.9
END
PARAM Z=35.,P=1.0
END

STOP
```

F. DATA FILE FOR EXAMPLE 4-3.

ALPHA	BETA	J
0.0000	0.00000	15.000
2.0000	0.10000	14.937
2.0000	0.30000	15.050
2.0000	0.50000	15.042
2.0000	0.70000	15.020
2.0000	0.90000	15.002
2.0000	1.00000	14.994
4.5000	0.10000	14.899
4.5000	0.30000	14.982
4.5000	0.50000	14.982
4.5000	0.70000	14.970
4.5000	0.90000	14.959
4.5000	1.00000	14.955
7.5000	0.10000	14.862
7.5000	0.30000	14.912
7.5000	0.50000	14.918
7.5000	0.90000	14.912
7.5000	1.00000	14.911
9.0000	0.10000	14.846
9.0000	0.30000	14.881
9.0000	0.50000	14.889
9.0000	0.70000	14.890
9.0000	0.90000	14.890
9.0000	1.00000	14.890
13.0000	0.10000	14.812
13.0000	0.30000	14.809
13.0000	0.50000	14.820
13.0000	0.70000	14.829
13.0000	0.90000	14.836
13.0000	1.00000	14.839
16.5000	0.10000	14.791

16.5000	0.30000	14.759
16.5000	0.50000	14.769
16.5000	0.70000	14.782
16.5000	0.90000	14.794
16.5000	1.00000	14.799
19.0000	0.10000	14.779
19.0000	0.30000	14.730
19.000	0.50000	14.737
19.000	0.70000	14.752
19.000	0.90000	14.766
19.000	1.00000	14.773
23.500	0.10000	14.765
23.500	0.30000	14.686
23.500	0.50000	14.689
23.500	0.70000	14.704
23.500	0.90000	14.722
23.500	1.00000	14.730
26.000	0.10000	14.760
26.000	0.30000	14.667
26.000	0.50000	14.666
26.000	0.70000	14.681
26.000	0.90000	14.699
26.000	1.00000	14.708
29.500	0.10000	14.755
29.500	0.30000	14.646
29.500	0.50000	14.638
29.500	0.70000	14.652
29.500	0.90000	14.671
29.500	1.00000	14.680
35.000	1.00000	14.754
35.000	0.30000	14.621
35.000	0.50000	14.603
35.000	0.70000	14.614
35.000	0.90000	14.632
35.000	1.00000	14.642

APPENDIX D. USER'S MANUAL

When we wish to plot the constant cost curves, the following described procedures should be performed as indicated. In this section, let us consider the specific problem in example 4-1.

1. PLOTTING THE PARAMETER PLANE CURVES.

The first step is to plot the parameter plane curves which are constant zeta curves and constant omega curves as shown in Figure 12 in Chapter 4. This curve permits us to choose the range of two parameters that are to be used in the analysis and design.

2. GENERATING A DATA FILE.

The second is to obtain a data file containing α , β and performance index values. Simulation in D.S.L is used to evaluate the cost at a large number of points on the parameter plane. The simulation results look as follows.

```
1     DSL/VS ..... RELEASE 1 ..... MODIFICATION LEVEL 1 ..... 1985


0
0                    >>>> DSL SIMULATION INPUT DATA <<<<
        TITLE PERFORMANCE INDEX CALCULATION
        TABLE Z(1)=100.,P(1-4)=1.0,11.0,10.0,0.0,D(1)=1.0
        PARAM A=0.0,B=0.0,Y0=0.0
        CONTROL FINTIM=15,DELT=0.01
        PRINT 15.,ALPHA,BETA,IES,IAET,IAE
        END
0***  KLENGH =  2189, KPOINT =   227, AVAILABLE SPACE LEFT IN COMMON/CURVAL/ =
0***  RK5     INTEGRATION METHOD USED ***
0        *** DSL OUTPUT LISTING, GROUP  1
+                                                    , STARTING RUN   1 ***
         PERFORMANCE INDEX CALCULATION
    TIME        ALPHA        BETA          IES          IAET          IAE
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
   15.000      0.00000E+00  0.00000E+00   15.000       112.50       15.000
0
0                    >>>> DSL SIMULATION INPUT DATA <<<<
        PARAM A=4.,B=10.
        END
0        *** DSL OUTPUT LISTING, GROUP  1
+                                                    , STARTING RUN   2 ***
         PERFORMANCE INDEX CALCULATION
    TIME        ALPHA        BETA          IES          IAET          IAE
  0.00000E+00   4.0000       10.000      0.00000E+00  0.00000E+00  0.00000E+00
   15.000       4.0000       10.000      0.32856      0.47462      0.64140
0
```

We have to edit the data file as shown in Appendix C to remove unnecessary parts.

3. LOADING A DATA FILE.

The third step is to download this file from the mainframe computer to a floppy diskette. This data now on the floppy diskette can be copied on the IBM microcomputer under a designated filename. In this example, the filename is called pi1.dat. Now, we are prepared to plot the families of performance index curves in the same range of parameter values on the two parameter plane.

4. PLOTTING THE COST CURVES.

To run the performance index program, type in the executable file name, COSTFUN. The IBM microcomputers in the NPS Controls Lab are set up under the file management system 1DIR. To execute the program under the this system, enter the appropriate subdirectory by using the up/down arrow keys to position the cursor adjacent to the subdirectory name, then depress the ENTER key. Use this same procedure to select COSTFUN.EXE and push ENTER to run the program. The preparation step for plotting the desired curves will appear.

Enter proper values after each prompt as follows:

```
1.How many constant PI curves # ===>6
2.Enter PI values ,lwgt,ldig #  1===>0.27,1,2
2.Enter PI values ,lwgt,ldig #  2===>0.2,1,1
2.Enter PI values ,lwgt,ldig #  3===>0.15,1,2
2.Enter PI values ,lwgt,ldig #  4===>0.12,1,2
2.Enter PI values ,lwgt,ldig #  5===>0.11,1,2
2.Enter PI values ,lwgt,ldig #  6===>0.1,1,1
3.Enter number of data file points ===>86
```

Next, adjust the plot size to the same parameter plane curves.

```
-----------------------------------------------------
*** This is the routine to Adjust the plot size  ***
-----------------------------------------------------


Output plots are currently sized to fill the console screen.

Would you like to adjust the plot size?
Enter "y" or "n" ==> y



A default plot factor size of 0.9 is used to fill the console screen.
This plot size will be halved by entering a value of 0.45.
It will be doubled with a factor entry of 1.8.

Enter factor size ==> 0.7
```

The last step is to enter a data file name.

```
-----------------------------------------------------
*** This is the routine to LOAD data from a file ***
-----------------------------------------------------

File name should not exceed 8 characters
File extension should not exceed 3 characters



What is the file name (fn) and extension (ext)?

Enter in form fn.ext (e.g. PARAPLAN.INP) ===>  pi1.dat
```

After displaying the graph on the monitor, the program prompts you for a hardcopy of the graph.

```
          WOULD YOU LIKE HARDCOPY OF THESE PLOTS? (Y or N)
       y


 -----------------------------------------------------------------
 ¦                     PRINTER/OUTPUT MENU                       ¦
 -----------------------------------------------------------------
 ¦  PRINTER NO.   ¦                PRINTER                       ¦
 -----------------------------------------------------------------
 ¦       1        ¦    Epson FX-80, All                          ¦
 ¦       2        ¦    Epson FX-100, All                         ¦
 ¦       3        ¦    Epson MX-100, All                         ¦
 ¦       4        ¦    Epson RX-80, All                          ¦
 ¦       5        ¦    Epson MX-80 & IBM Printer                 ¦
 ¦       6        ¦    HP 7470A Graphics Plotter                 ¦
 ¦       7        ¦    HP 7475A Graphics Plotter                 ¦
 ¦       8        ¦    HP 758xB Series Plotters                  ¦
 ¦       9        ¦    HP 2686A Laser Jet                        ¦
 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 ¦      10        ¦    Graphics Monitor (default)                ¦
 ¦      11        ¦    HARDWARE Interface Menu                   ¦
 ¦      12        ¦    Input PLOT88 Values for IOPORT and MODEL  ¦
 ¦      99        ¦    RETURN to LOAD Data                       ¦
 -----------------------------------------------------------------


       Enter integer number for selection ===>  5
```

Reenter the data file name for the hard copy. Figure 13 is a performance index curve for constraint $\int E^2\,dt$. In like manner, the other two nonlinear criterion of performance index in terms of $\int |E|\, t\, dt$ and $\int |E|\, dt$ can also be plotted by running the same program. Figure 14 and Figure 15 are shown for desired constraint $\int |E|\, t\, dt$ and $\int |E|\, dt$, respectively.

LIST OF REFERENCES

1. Newton, G.C., Gould, L.A. and Kaiser, J.F., *Analytical Design of Linear Feedback Controls,* John Wiley and Sons, New York, 1957.

2. I.B.M., *Dynamic Simulation Language,* 1987.

3. Young, T.L. and Van Woert, M.L., *PLOT88 Software Library Reference Manual,* Plotworks Inc., January 1984.

4. Thaler, G.J., "Linear Control Design (ECE-4320)," Unpublished Draft.

5. Reid, R.E. and Moore, J.W., *Optimal Steering Control of High Speed Containership, Part I System Dynamics, Part II Controller Design,* Proc 1980 ACC, San Francisco, CA, August 1980

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor George J. Thaler, Code 62Tr Naval Postgraduate School Monterey, CA 93943-5000	5
5.	Professor Harold A. Titus, Code 62Ts Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Lcdr R.J Kranz 1740 Dietz Place NW Albuquerque, NM 87107	1
7.	Captain Roger Nutting, Director Ship Design Group, Code SEA-50 NAVAL SEA SYSTEMS COMMAND Washington, D.C. 20362-5101	1
8.	Mr. Dean Carico, Code RW-042 RWATD NATCL SEA SYSTEMS COMMAND Patuxent River, MD 20670	1
9.	Library of the Naval Academy Anggok Dong, Jinhae city, Gyungnam 602-00 Republic of Korea	1
10.	Shin, Dong Ryong 3-317 18/4 Bo Kwang dong, Young San Gu, Seoul city 140-00 Republic of Korea	10
11.	Naval Shipyard Hyun dong Jinhae city, Gyungnam 602-00 Republic of Korea	1

12. Lcdr Hwang, Jung Sub 1
SMC1209 Naval Postgraduate School
Monterey, CA 93943

13. Lt Choi, Man Soo 1
SMC1840 Naval Postgraduate School
Monterey, CA 93943

14. Lt Yoon, Hee Byung 1
SMC2137 Naval Postgraduate School
Monterey, CA 93943

15. Lt Hur, Hong Bum 1
SMC2061 Naval Postgraduate School
Monterey, CA 93943

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

323 - 899

Thesis
S4741 Shin
c.1 Parameter plane design
method.

Thesis
S4741 Shin
c.1 Parameter plane design
method.

thesS4741
Parameter plane design method.
3 2768 000 81875 1
DUDLEY KNOX LIBRARY